

CA-7[®]

Database Maintenance Guide

3.3

This documentation and related computer software program (hereinafter referred to as the "Documentation") is for the end user's informational purposes only and is subject to change or withdrawal by Computer Associates International, Inc. ("CA") at any time.

THIS DOCUMENTATION MAY NOT BE COPIED, TRANSFERRED, REPRODUCED, DISCLOSED, OR DUPLICATED, IN WHOLE OR IN PART, WITHOUT THE PRIOR WRITTEN CONSENT OF CA. THIS DOCUMENTATION IS PROPRIETARY INFORMATION OF CA AND PROTECTED BY THE COPYRIGHT LAWS OF THE UNITED STATES AND INTERNATIONAL TREATIES.

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 THE END USER OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION, LOST PROFITS, BUSINESS INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF CA IS EXPRESSLY ADVISED OF SUCH LOSS OR DAMAGE.

THE USE OF ANY PRODUCT REFERENCED IN THIS DOCUMENTATION AND THIS DOCUMENTATION IS GOVERNED BY THE END USER'S APPLICABLE LICENSE AGREEMENT.

The manufacturer of this documentation is Computer Associates International, Inc.

Provided with "Restricted Rights" as set forth in 48 C.F.R. Section 12.212, 48 C.F.R. Sections 52.227-19(c)(1) and (2) or DFARS Section 252.227.7013(c)(1)(ii) or applicable successor provisions.

First Edition, September 2000

© 1988-2000 Computer Associates International, Inc.
One Computer Associates Plaza, Islandia, NY 11749
All rights reserved.

All trademarks, trade names, service marks, or logos referenced herein belong to their respective companies.

Contents

Chapter 1. Introduction	1-1
1.1 Summary of Revisions	1-2
1.1.1 Product Changes	1-2
1.1.2 Documentation Changes	1-5
1.2 CA-7 Overview	1-6
1.3 Functional Overview	1-7
1.3.1 Online Assistance	1-7
1.3.1.1 HELP Facility	1-7
1.3.1.2 PF Key Assignment	1-7
1.3.2 Workload Scheduling	1-8
1.3.3 Workload Sequencing	1-8
1.3.4 Work Flow Control	1-8
1.3.5 Virtual Resource Management	1-8
1.3.6 Automated Recovery Facility	1-9
1.3.7 Job Restart	1-10
1.3.8 Online Utility Execution	1-10
1.3.9 Security	1-10
1.3.10 JCL Overrides	1-11
1.3.11 External Communications Facilities	1-11
1.3.12 Batch Card Load Program	1-11
1.3.13 Workload Forecasting	1-11
1.3.14 Workload Planning	1-12
1.3.15 Workload Balancing	1-12
1.3.16 Workload Documentation	1-12
1.3.17 Management Level Reporting	1-13
1.3.17.1 Automated Performance Analysis (APA) Reporting	1-13
1.3.17.2 History Reporting	1-13
1.3.17.3 CA-Earl Reporting	1-13
1.3.17.4 CA-Easytrieve Plus Reporting	1-13
1.3.18 CA-7 Text Editor	1-13
1.4 Online Input	1-14
1.4.1 PF Keys	1-14
1.4.2 Formatted Screen Messages	1-15
1.4.3 CA-7 Function Menu	1-16
1.4.4 DB Menu Screen	1-17
1.4.4.1 Usage Notes	1-17
1.4.5 Bypassing the Menu	1-18
1.4.5.1 Bypassing Secondary Menus	1-18
1.4.6 Function Shortcuts	1-19
1.4.6.1 Shortcut Examples	1-19
1.4.7 Function Transfer	1-20
1.4.8 Aliases for Formatted Screen Functions	1-20
1.5 Batch Input	1-22
1.5.1 Function Shortcuts and Bypassing the Menu	1-22
1.5.2 Function Transfer	1-23
1.5.3 Aliases for Formatted Screen Functions	1-23

Chapter 2. Jobs	2-1
2.1 Adding a Job	2-2
2.2 DB.1 - CPU Job Definition Screen	2-3
2.2.1 Field Descriptions	2-4
2.2.2 Usage Notes	2-15
2.2.3 Batch Input Example	2-15
2.3 Changing a Job	2-16
 Chapter 3. Scheduling	 3-1
3.1 Work Scheduling	3-2
3.1.1 Date and Time Scheduling	3-3
3.1.1.1 Base Calendars	3-3
3.1.1.2 Schedule Definition	3-4
3.1.1.3 Schedule Resolution	3-4
3.1.2 Schedule Modifications	3-5
3.1.2.1 Schedule Scan	3-6
3.1.3 Event Scheduling	3-6
3.1.3.1 Triggers	3-6
3.1.4 On-Request Scheduling	3-7
3.1.4.1 DEMAND Command	3-7
3.1.4.2 DMDNW Command	3-8
3.1.4.3 RUN Command	3-8
3.1.4.4 RUNNW Command	3-8
3.2 Defining Schedules	3-9
3.2.1 DB.2 - Scheduling Menu Screen	3-9
3.2.1.1 Usage Notes	3-10
3.2.2 DB.2.1 - CPU Job Scheduling Screen	3-11
3.2.2.1 Field Descriptions	3-12
3.2.2.2 Usage Notes	3-13
3.2.3 CPU Job Scheduling Parameter Edit Screen	3-14
3.2.3.1 Field Descriptions	3-15
3.2.3.2 Batch Input Example	3-23
3.2.4 DB.2.2 - Input Network Scheduling Screen	3-24
3.2.4.1 Field Descriptions	3-25
3.2.5 Input Network Scheduling Parameter Edit Screen	3-27
3.2.5.1 Field Descriptions	3-28
3.2.5.2 Batch Input Example	3-35
3.2.6 DB.2.3 - Output Network Scheduling Screen	3-36
3.2.6.1 Field Descriptions	3-37
3.2.6.2 Usage Notes	3-38
3.2.7 Output Network Scheduling Parameter Edit Screen	3-39
3.2.7.1 Field Descriptions	3-40
3.2.7.2 Batch Input Example	3-42
3.2.8 Trigger Scheduling Screens	3-43
3.2.9 DB.2.4 - Job Triggering Screen	3-44
3.2.9.1 Usage Notes	3-44
3.2.10 DB.2.5 - Input Network Triggering Screen	3-45
3.2.10.1 Usage Notes	3-45
3.2.11 DB.2.6 - Data Set Triggering Screen	3-46
3.2.11.1 Usage Notes	3-46

3.2.11.2	Field Descriptions	3-47
3.2.11.3	Batch Input Example	3-51
3.2.12	DB.2.7 - Modification to Resolved Schedule Dates Screen	3-52
3.2.12.1	Field Descriptions	3-53
3.2.12.2	Usage Notes	3-54
3.2.13	DB.2.8 - Base Calendar Maintenance	3-55
3.2.13.1	Field Descriptions	3-56
3.2.13.2	Usage Notes	3-58
Chapter 4.	Requirement Definitions	4-1
4.1	Execution Requirements and the LOAD Process	4-2
4.1.1	Dependence Definition	4-2
4.1.2	Database Definitions	4-2
4.1.3	Data Set Dependencies	4-3
4.1.4	Other Dependencies	4-3
4.1.5	Temporary Predecessors	4-4
4.2	Satisfying Requirements	4-5
4.3	Defining Predecessors and Successors	4-6
4.3.1	DB.3 - Job Predecessor/Successor Menu Screen	4-6
4.3.1.1	Usage Notes	4-7
4.3.2	DB.3.1 - Data Set Predecessors Screen	4-8
4.3.2.1	Field Descriptions	4-9
4.3.2.2	Usage Notes	4-12
4.3.2.3	Batch Input Example	4-12
4.3.3	DB.3.2 - CPU Job Predecessors Screen	4-13
4.3.3.1	Field Descriptions	4-14
4.3.3.2	Usage Notes	4-18
4.3.3.3	Batch Input Example	4-18
4.3.4	DB.3.4 - Input/Output Network Tasks Screen	4-19
4.3.4.1	Field Descriptions	4-20
4.3.4.2	Usage Notes	4-24
4.3.4.3	Batch Input Example	4-24
4.3.5	DB.3.6 - User Memo-Form Predecessors Screen	4-25
4.3.5.1	Field Descriptions	4-26
4.3.5.2	Usage Notes	4-28
4.3.5.3	Batch Input Example	4-28
4.3.6	DB.3.7 - Report IDs Created Screen	4-29
4.3.6.1	Field Descriptions	4-30
4.3.6.2	Batch Input Example	4-31
Chapter 5.	Virtual Resource Management	5-1
5.1	Resource Types	5-2
5.1.1	Shared Resources	5-2
5.1.2	Exclusive Resource	5-2
5.1.3	Corequisite Resources	5-2
5.1.4	Address Space Resources	5-2
5.1.5	Resource Count Resources	5-3
5.2	Handling Resource Conflicts	5-4
5.3	VRM Menu Screen	5-5
5.3.1	Usage Notes	5-5

5.4 RM.1 Job Resource Management Screen	5-6
5.4.1 Field Descriptions	5-7
5.4.2 Batch Input Example	5-10
5.4.3 Critical Path Monitoring Corequisite Resources	5-10
5.4.3.1 Example	5-11
5.4.3.2 Notes	5-11
5.5 RM.2 Job/Resource Cross Reference List Screen	5-12
5.5.1 Field Descriptions	5-13
5.5.2 Batch Input Example	5-14
5.6 RM.3 Active Job Resources Display Screen	5-15
5.6.1 Field Descriptions	5-16
5.6.2 Batch Input Example	5-17
5.7 RM.4 Pending Resources Job Display Screen	5-18
5.7.1 Field Descriptions	5-19
5.7.2 Batch Input Example	5-19
5.8 RM.5 Jobs Waiting on Resources Screen	5-20
5.8.1 Field Descriptions	5-21
5.8.2 Batch Input Example	5-22
5.9 RM.6 Corequisite Resources List Screen	5-23
5.9.1 Field Descriptions	5-24
5.9.2 Batch Input Example	5-24
5.10 RM.7 Resource Count Resource Management Screen	5-25
5.10.1 Field Descriptions	5-26
5.10.2 Batch Input Example	5-27
5.11 VRM Device Control	5-28
5.11.1 Overview	5-28
5.11.2 VRM Device Control Functions - Definition Structure	5-28
5.11.3 VRM Device Control Functions - Definition Control	5-29
5.11.4 VRM Device Control Functions - Submission Control	5-31
5.11.5 VRM Device Control - Activation	5-31
 Chapter 6. Automated Recovery Facility	 6-1
6.1 Monitoring Exception Conditions	6-2
6.2 Responding to Exception Conditions	6-3
6.3 Types of Exceptions Recognized by ARF	6-4
6.4 ARFSET Structure	6-6
6.4.1 ARF Definition Structure	6-6
6.4.2 Filter Criteria	6-6
6.4.3 Type Specific Tests	6-7
6.4.4 Responses	6-7
6.4.4.1 Types of Responses	6-8
6.4.5 Final Disposition	6-8
6.5 Implementation Considerations	6-9
6.6 AR.3 ARF Condition Definition Maintenance Screen	6-10
6.6.1 Field Descriptions	6-10
6.7 ARF Condition Definition Edit Screen	6-12
6.7.1 Field Descriptions	6-13
6.7.2 Field Descriptions - Filter Criteria	6-15
6.7.3 Field Descriptions - Type Specific Tests for JC and SC Conditions	6-19

6.7.4	Field Descriptions - Type Specific Tests for EC, EE, IS, LB, LE and LS	
	Conditions	6-23
6.7.5	Field Descriptions - Responses	6-27
6.7.6	Field Descriptions - Final Disposition	6-28
6.8	Rules for Coding ARF Action Statements	6-31
6.8.1	AC - Issue a Command	6-31
	6.8.1.1 Example	6-31
6.8.2	AM - Issue a Message	6-32
	6.8.2.1 Examples	6-33
6.8.3	AW - Wait	6-34
	6.8.3.1 Example	6-34
6.8.4	AJ - Schedule a Recovery Job	6-35
	6.8.4.1 Example	6-36
6.9	Using Reserved Words in Type Specific Tests	6-37
6.10	Using Variables in ARF Action Statements	6-39
6.11	Examples of ARF Condition Definition	6-40
	6.11.1 Defining a Job Completion Condition (JC)	6-40
	6.11.2 Defining a Late at Job End Notification (LE)	6-42
Chapter 7.	Workload Documentation	7-1
7.1	Workload Documentation Screens	7-2
7.1.1	DB.4 Menu Screen	7-3
	7.1.1.1 Usage Notes	7-3
7.1.2	DB.4.1 - CPU Job Documentation Screen	7-4
	7.1.2.1 Field Descriptions	7-5
7.1.3	DB.4.2 - Input/Output Network Documentation Screen	7-7
	7.1.3.1 Field Descriptions	7-8
7.1.4	DB.4.3 - User-Defined Item Documentation Screen	7-11
	7.1.4.1 Field Descriptions	7-12
7.1.5	DB.4.4 - Data Set Documentation Screen	7-15
	7.1.5.1 Field Descriptions	7-16
7.1.6	DB.4.5 - DD Statement Documentation Screen	7-19
	7.1.6.1 Field Descriptions	7-20
7.1.7	DB.4.6 - Application System Documentation Screen	7-23
	7.1.7.1 Field Descriptions	7-24
7.2	Special Documentation Features	7-26
7.2.1	User-Defined Documentation Segments	7-26
	7.2.1.1 Reserved Segment Names	7-26
	7.2.1.2 Rules and Guidelines for Documentation Segmentation	7-26
	7.2.1.3 Defining Segment Names	7-27
	7.2.1.4 Format of Documentation Segments	7-27
	7.2.1.5 Displaying Documentation Segments	7-28
	7.2.1.6 Subsegments	7-29
	7.2.1.7 Rules and Guidelines for Subsegments	7-29
	7.2.1.8 Defining Subsegment Names	7-29
	7.2.1.9 Format of Documentation Subsegments	7-30
	7.2.1.10 Displaying Documentation Subsegments	7-31
	7.2.1.11 Documenting Naming Conventions	7-32
7.2.2	Special Purpose Documentation	7-33
	7.2.2.1 General Format	7-33

7.2.3	#QDESC Segment Usage	7-34
7.2.4	#station-name Segment Usage	7-35
7.2.5	#RESTART Segment Usage	7-36
7.2.6	#END Statement	7-37
Chapter 8. Workstation Networks		8-1
8.1	Adding a Network	8-2
8.2	DB.5 - Input/Output Network Definition Screen	8-3
8.2.1.1	Field Descriptions	8-4
8.2.1.2	Batch Input Example	8-5
8.3	Changing a Network	8-6
Chapter 9. Data Sets		9-1
9.1	Adding a Data Set	9-2
9.1.1	Permanent Data Sets	9-3
9.1.2	Dynamically Allocated Data Sets	9-4
9.1.3	External Data Sets	9-4
9.2	DB.6 - Data Set Definition Screen	9-5
9.2.1.1	Field Descriptions	9-6
9.2.1.2	Usage Notes	9-9
9.2.1.3	Batch Input Example	9-9
9.3	Changing a Data Set	9-10
Chapter 10. JCL Management		10-1
10.1	DB.7 - JCL Library Maintenance Screen	10-2
10.1.1	Field Descriptions	10-3
10.1.2	Usage Notes	10-5
10.2	Scheduled Overrides	10-6
10.2.1	#JI and #JO Statements	10-7
10.2.1.1	Syntax	10-7
10.2.2	#JEND Statement	10-10
10.2.2.1	Syntax	10-10
10.2.2.2	Usage Notes	10-10
10.2.3	#XI, #XO, and #XEND Statements	10-10
10.2.4	Scheduled Override Examples	10-11
10.3	Additional Override Statements	10-13
10.3.1	#ARF Statement	10-14
10.3.1.1	Syntax	10-14
10.3.1.2	Example	10-14
10.3.1.3	Usage Notes	10-15
10.3.2	#MSG Statement	10-16
10.3.2.1	Syntax	10-16
10.3.2.2	Usage Notes	10-16
10.3.3	#RES Statement	10-17
10.3.3.1	Syntax	10-17
10.3.4	#SCC Statement	10-18
10.3.4.1	Syntax	10-18
10.3.4.2	Example	10-19
10.3.4.3	Usage Notes	10-20
10.3.5	Additional Override Examples	10-21

10.4	SASSJCLU - JCL Utility	10-24
10.4.1	Usage Notes	10-25
10.4.1.1	Control Statement	10-25
10.4.1.2	JCL	10-26
10.5	JCL Validation	10-27
10.6	LOAD Command Processing	10-28
10.7	Special Override Library	10-29
10.7.1	Defining a Special Override Library	10-29
10.7.2	Creating Temporary JCL	10-29
10.7.3	Using Temporary JCL	10-30
10.7.4	Other Considerations	10-30
10.8	Alternate JCL Libraries	10-31
10.8.1	Defining an Alternate Library	10-31
10.8.2	Creating Temporary JCL	10-31
10.8.3	Using Temporary JCL	10-31
10.8.4	Other Considerations	10-32
Chapter 11.	Edit Facility	11-1
11.1	Active Area	11-3
11.2	Edit Work File	11-4
11.3	Environments	11-5
11.4	Using the Editor	11-6
11.4.1	Invoking the Editor	11-6
11.4.2	Editing Text Data	11-6
11.4.3	Leaving the Editor	11-7
11.5	CA-7 Text Editor Environment	11-8
11.5.1	Edit Modes	11-8
11.5.1.1	Usage Considerations	11-8
11.5.1.2	Full Edit Mode (FEM)	11-9
11.5.2	Updating Text	11-27
11.5.3	Creating Text	11-28
11.5.4	Special Considerations	11-29
11.5.4.1	PF/PA Key Usage	11-29
11.5.4.2	Character Translation	11-29
11.5.4.3	Nondisplayable Data (hex)	11-29
11.6	TSO/ISPF Editor Environment	11-30
11.6.1	Requesting CA-7 Edit Functions from the ISPF Editor	11-31
11.6.2	Default Requests for CA-7 Edit Functions	11-32
11.6.3	Special Considerations	11-34
11.6.3.1	ISPF Edit Profile Settings	11-34
11.6.3.2	PF/PA Key Usage	11-34
11.6.3.3	SUBMIT Function	11-34
11.6.3.4	Size of Data	11-34
Chapter 12.	Database Verification	12-1
12.1	Pointer Verification	12-2
12.1.1	Pointer Verification Parameters	12-2
12.1.2	Pointer Verification by Record Type	12-3
12.1.2.1	Network Directory	12-3
12.1.2.2	Job Schedule Directory	12-3

12.1.2.3	Data Set Directory	12-4
12.1.2.4	Documentation Directory	12-4
12.1.2.5	Job Directory	12-4
12.1.2.6	Data Set Member	12-4
12.1.2.7	Network Member	12-5
12.1.2.8	Input Network Schedule Member	12-5
12.1.2.9	Output Network Schedule Member	12-5
12.1.2.10	Documentation Member	12-5
12.1.2.11	Job Member	12-5
12.2	Analyze Commands	12-6
Chapter 13.	Database Transportability	13-1
13.1	Assumptions	13-3
13.1.1	LOAD/RELOAD Status	13-4
13.1.2	CA-11 Step Insertion	13-4
13.1.3	Base Calendars	13-4
13.1.4	Calendar Schedules	13-4
13.1.5	DB.2.7 Schedule Modifications	13-4
13.1.6	NXTCYC Settings	13-5
13.1.7	Execution JCL Libraries	13-5
13.1.8	Cataloged Procedures	13-5
13.1.9	In-stream JCL Procedures	13-6
13.1.10	Workstation Networks	13-6
13.1.11	User Level Documentation	13-6
13.1.12	ANALYZE Application	13-6
13.1.13	Commas in the Data	13-6
13.1.14	Sequence Numbers	13-7
13.1.15	User ID Security	13-7
13.1.16	LINKed Documentation	13-7
13.2	Database Extraction	13-8
13.3	CPU Jobs	13-13
13.3.1	Adding at New Site	13-13
13.3.2	Requirements and Network Connections	13-15
13.3.2.1	JOB Connections	13-15
13.3.2.2	NWK Connections	13-15
13.3.2.3	DSN Connections	13-16
13.3.2.4	USR Connections	13-16
13.3.3	Deleting at Original Site	13-17
13.3.4	Disabling at Original Site	13-18
13.4	Workstation Networks	13-19
13.4.1	Adding at New Site	13-19
13.4.2	Deleting at Original Site	13-20
13.4.3	Disabling at Original Site	13-21
13.4.3.1	Input Workstation Networks	13-21
13.4.3.2	Output Workstation Networks	13-21
13.4.4	Unconnected Workstation Networks	13-21
13.5	Application Data Sets	13-22
13.5.1	Adding at New Site	13-22
13.5.2	Deleting at Original Site	13-23
13.6	PDS JCL Members	13-24

13.6.1 Adding at New Site	13-24
13.6.2 Deleting at Original Site	13-25
13.7 CA-Librarian JCL	13-26
13.7.1 Adding at New Site	13-26
13.7.2 Deleting at Original Site	13-28
13.8 CA-Panvalet JCL	13-29
13.8.1 Adding at New Site	13-29
13.8.2 Disabling at Original Site	13-31
13.9 Cataloged Procedures	13-32
13.9.1 Adding at New Site	13-32
13.9.2 Deleting at Original Site	13-33
13.10 Documentation	13-34
13.10.1 Adding at New Site	13-34
13.10.2 Deleting at Original Site	13-35
13.11 Schedules	13-36
13.11.1 Adding at New Site	13-36
13.11.1.1 CPU Job Schedules	13-36
13.11.1.2 Job Triggers	13-37
13.11.1.3 Network Triggers	13-37
13.11.1.4 Data Set Triggers	13-37
13.11.1.5 Input Networks	13-38
13.11.1.6 Output Networks	13-39
13.11.2 Deleting at Original Site	13-40
13.11.2.1 CPU Job Schedules	13-40
13.11.2.2 Input Network Schedules	13-40
13.11.2.3 Output Network Schedules	13-40
13.12 Creating Command Files	13-41
13.12.1 Job 1	13-41
13.12.1.1 PARM Keywords	13-41
13.12.1.2 Specifying Jobs to Move	13-42
13.12.1.3 Data Sets Used/Created	13-43
13.12.2 Job 2	13-44
13.12.2.1 PARM Keywords	13-44
13.12.2.2 Data Sets Used/Created	13-44
13.12.3 Job 3	13-46
13.12.3.1 PARM Keywords	13-46
13.12.3.2 Command Data Sets	13-47
13.13 Special Considerations	13-50
13.13.1 Design Limitations	13-50
13.13.2 Adding to the New Database	13-51
13.13.3 Running at New Site	13-51
13.13.4 Mass Changes at Existing Site	13-52
13.14 Virtual Resource Management Database Extracts	13-53
13.14.1 VRM DBT Job Extract JCL	13-54
13.14.1.1 PARM Keywords	13-54
13.14.1.2 SYSIN Control Cards	13-55
13.14.1.3 Data Sets Used/Created	13-55
13.15 Automated Recovery Facility Database Extracts	13-56
13.15.1 PARM Keywords	13-57
13.15.2 Data Sets Used/Created	13-58

Index	X-1
------------------------	------------

Chapter 1. Introduction

The *CA-7 Database Maintenance Guide* is intended for database administrators and/or change control personnel. It contains information on:

- jobs
- scheduling
- requirement definitions
- virtual resource management
- automated recovery facility
- workload documentation maintenance
- networks
- data sets
- JCL management
- text editing
- database verification
- database transportability

1.1 Summary of Revisions

This topic explains changes to both CA-7 and to the documentation.

1.1.1 Product Changes

CA-7 Version 3.3 contains the following major enhancements:

- Parallel Sysplex Exploitation

CA-7 can optionally maintain a memory structure in the Coupling Facility in which participating ICOMs record tracking data. One or more Host ICOM(s) read from the memory structure and write to the Communication data set. This can significantly reduce I/O contention and increase feedback throughput.

- UNIX System Services Interface

The OS/390 UNIX System Services (USS) CA-7 interface allows communication with CA-7 from the USS environment. The interface can be called directly from the UNIX shell or from the IBM USS batch interface (BPXBATCH).

- CA-7 CCI Interface

The CA-7 CCI interface allows two-way communication with CA-7 from other address spaces and environments. The interface can be engaged in a batch mode, in a REXX address environment or it can be called directly from a user program. It accepts single or stacked commands as input and returns the CA-7 output from the commands as if they had been executed in batch mode.

- Critical Path Monitoring

Through integration with CA-OPS/MVS II, Unicenter TNG and Unicenter TNG MVS Event Manager Option (MEMO), CA-7 can support the definition and monitoring of critical job flows within the CA-7 workload. CA-OPS/MVS II provides management and administration of critical path displays.

- Mixed Case Support in CA-7 Editor

Character translation controls can be set in the CA-7 Editor. New Editor subcommands 'UPPER' and 'MIXED' determine whether editor data is translated to uppercase or left "as is."

These subcommands are enabled with a new initialization file option. If this option is not coded, then all edit data is translated to uppercase.

- Job Completion Tracking Precision

CA-7 records job completion times in hundredths of seconds. This allows job completions to be discriminated with a high degree of precision, thus reducing the likelihood of requirement posting ambiguities where jobs complete within the same minute.

- Display Duplicate Days for RESOLVE

CA-7 can optionally display the duplicate RESOLVE day(s) in new message SRC1-137. This occurs when a job is scheduled to execute the same day under two or more different Schedule IDs. With this information one can more quickly and efficiently determine the source of the scheduling conflict.

- VRM Device Control

Virtual Resource Management (VRM) Device Control provides an alternative to Workload Balancing control of job submission based on tape drive availability. VRM resource count resources representing the number and type of storage devices used by the job are defined dynamically during CA-7 LOAD processing.

Workload Balancing only permits two types of tape drives. With VRM Device Control, the number and structure of device groups is determined by the user.

- CA-7 Command Retrieval

Command line input for CA-7 VTAM terminals is recorded in storage and may be retrieved with the /FETCH command. When the /PFnn command is used to associate /FETCH with a PF key, the CA-7 user can conveniently retrieve the last five CA-7 commands entered at an online terminal.

- CA-7 Base Calendar Security

CA-7 security can allow clients to define CA-7 base calendar names to an external security product and secure user access to individual base calendars.

- REXX Address Environment

Using the new CA-7 CCI interface, CA-7 allows REXX programs to pass commands to CA-7 and take action based on the output from those commands.

- Job 'Purge' Function

The DB.1 (Job) panel provides a new function, PURGE, which deletes all CA-7 database records related to a job. In addition to the standard delete processes, the PURGE function deletes incoming trigger definitions, requirement successor definitions, and the CA-11 CMT member for the job.

- Suppress LATE Designation

Through an Initialization File option, the PROMPTS field on the DB.1 (Job) panel can be used to indicate certain jobs should never be marked as LATE on status displays. This means operations and production control staff will not be distracted when test or non-critical jobs do not complete on time.

- CSA Chains Above the 16M Line

CA-7 CSA SMF and Trailer chains now reside in extended CSA (above-the-line), thereby reducing utilization of this critical resource.

- Automated Recovery Facility (ARF) Enhancements

CA-7 can optionally add a LOGON parameter to the ARF TSO SEND command to cause messages to be retained until the user logs on to TSO. Also, support for ARF has been added to the Database Transportability facility.

- Prior Run Queue Expansion

The maximum size of the Prior Run Queue is now approximately twice as large as in prior releases.

- CA-7 JCLCheck Common Component

The CA-JCLCheck Common Component is provided in place of the CA-7 JCL syntax checker.

- Documentation Files on Tape

The current CA-7 documentation files are provided in IBM Book Manager and PDF format on the product tape.

- Other Enhancements:

- SMF Purge records may optionally be sent to a test copy of CA-7. This allows detection of pre-execution JCL Errors by the test copy.
- The Scratch and Disk Queue Table queues can be formatted during a CA-7 ERST start which facilitates use of VIO to improve performance.
- The LJOB command provides a new option, LIST=RQEXCP, that lists only those requirements with a SKIP or ONLY indication.
- The reverse forecast commands, FRJOB and FRQJOB, have a new option, LIST=HDRS. This will limit the display to only the target job and all 'header' jobs.
- Database Transportability now supports a new keyword, NODSNS, for SASSDT30 which prevents the generation of data set definitions.
- The LQ family of commands (LREQ, LRDY, LACT, and so forth) now support a Schedule ID filter, SCHID=.
- The LRLOG command has a new sequence option, SEQ=REV, which causes entries to be displayed in reverse date/time sequence (most recent first).
- The OPTIONS initialization file statement has a new keyword DPROCCOM= to enable comment statements in CA-Driver procedures.
- The OPTIONS initialization file statement has a new keyword EXTSCHID= to set a default schedule ID for externally tracked jobs that are not assigned a non-zero schedule ID from the SASSEXTT table.
- The CA-7 CAIRIM initialization module now accepts a new reinitialization parameter (REINIT=UTABS) to reload only user defined table modules.
- The /DISPLAY command has a new STATUS option (/DISPLAY,ST=CA7) to describe the current copy of CA-7 (VTAM application ID and so forth).

1.1.2 Documentation Changes

The documentation for CA-7 Version 3.3 differs from previous releases as follows:

- The documentation set has been engineered to take advantage of the latest technology for online viewing, keyword searching, book marking, and printing. The set consists of a hard copy *CA-7 Getting Started* guide and Version 3.3 of CA-7 for OS/390 documentation in both IBM BookManager and Adobe Acrobat Reader format on the tape.
- Unicenter TNG Framework for OS/390 is composed of the services formerly known as CA90s and Unicenter TNG Framework.
- Reading Syntax Diagrams in the *CA-7 Commands Guide* explains how to read the command syntax used in all guides.

Technical changes are identified by a revision bar (|) in the left margin. Revision bars are not used for editorial changes and new manuals.

1.2 CA-7 Overview

CA-7 is a comprehensive Automated Production Control system. It has the capability to address the broad range of activities traditionally considered the responsibility of computer operation's production control. CA-7 is an online, realtime, interactive system which automatically controls, schedules and initiates work according to time-driven and/or event-driven activities.

1.3 Functional Overview

CA-7 capabilities and facilities include, but are not limited to, the following topics.

1.3.1 Online Assistance

CA-7 incorporates facilities to assist in using the system to control the production environment, to operate the system, and to retrieve information from the system.

1.3.1.1 HELP Facility

The HELP facility is an online function available at any CA-7 terminal. Information describing CA-7 features, applications and commands can be rapidly retrieved and displayed using this feature. The HELP command accesses an online HELP library and displays a general discussion of the requested application or command. Information concerning commands includes a brief description of the command, its keywords, formats, and uses.

1.3.1.2 PF Key Assignment

Program function (PF) keys may be assigned to define a specific command for a predetermined function. This enables the terminal user to enter common functions with a single keystroke per function. The program access (PA) keys can be used in a similar manner.

1.3.2 Workload Scheduling

CA-7 can schedule all activities associated with the production workload. In general, these time-driven and event-driven facilities are used to schedule workstation activities (for example, data entry, production control, mail room) and CPU jobs. However, these same facilities can be used to schedule any definable activity, whether or not that activity is related to the production workload or to CPU job processing.

CA-7 also has the ability to track batch jobs and started tasks that are submitted outside of CA-7 and data sets created by such jobs. The tracking is selective based on tables that are coded to specify the job names and/or data set names which are to be tracked. This is only available for CPUs that share the Communication data set with CA-7.

1.3.3 Workload Sequencing

Workload sequencing refers to facilities which prevent jobs from being executed before input tasks are successfully completed and job dependencies (successful input data set creation, predecessor job or network completion) are satisfied.

1.3.4 Work Flow Control

Although CA-7 automatically schedules and invokes a defined workload, sometimes it is necessary to circumvent scheduled work flow for new priorities. With CA-7, unscheduled interruptions are handled online so that revised priorities can be addressed immediately. Schedules can be moved forward and backward. Jobs and activities can be held, rushed or canceled online without time-consuming rescheduling activity.

Online prompting reminds appropriate workstations when scheduled work is late or in danger of becoming late. Prompting promotes work flow by drawing attention to required action on activities which might otherwise be overlooked or delayed.

1.3.5 Virtual Resource Management

The Virtual Resource Management facility (VRM) provides a mechanism to control job submission based on job resource use. A job to resource connection defines job usage and disposition of the resource and thereby allows an additional level of production workload control. Resource use is managed internally by CA-7 during the job submission and job/step completion processes and may be used to serialize a production cycle or critical resource access.

1.3.6 Automated Recovery Facility

The Automated Recovery Facility (ARF) for CA-7 provides the capability to closely monitor exception conditions for production jobs and to schedule recovery actions to execute at or near the point of failure.

One or more ARF definitions may provide the criteria used to evaluate the exception conditions for a job. A named set of ARF definitions that provide the criteria used to determine the exception conditions for a job is referred to as an ARFSET. ARF determines exception conditions based on criteria in the ARFSET that is named on the DB.1 panel for the job.

ARF specifically tests for the exception conditions that are described in the ARFSET for the job. Each ARFSET may contain up to 20 distinct definitions. Each definition in an ARFSET includes a description of an exception condition along with a set of responses that are to be executed when the exception is detected.

ARFSET definitions may specify criteria that are precisely tailored to fit exact exception conditions for an individual job or may be coded to apply to a broad range of exception conditions for any job.

Types of exception conditions monitored by ARF include, but are not limited to: late conditions, step or job completion errors (including tests for condition code errors, system and user abends) and elapsed time checking during execution or at job completion.

ARF can respond to exception conditions by issuing CA-7 or MVS commands. ARF can also submit and track special recovery jobs that are not part of the "normal" schedule. Special messages may be routed to TSO users or to the MVS console as part of ARF response processing. ARF can cause a job to be restarted, canceled or "force completed" as part of processing a job completion exception.

ARF attempts to respond to exception conditions at or near the time they are detected. For example, ARF can issue CA-7 or MVS commands immediately at the point in time when ARF detects a step completion code exception.

1.3.7 Job Restart

Under CA-7, jobs which abnormally terminate are automatically suspended and notification is made to a designated workstation advising of a need to perform a job restart. Through online commands, all jobs waiting for restart may be listed with job restart information. Job restart information identifies the last step successfully executed, the abend code, the restartable steps and, optionally, specific user-defined special instructions. Abended jobs can be restarted online after restart cleanup is completed.

If CA-11 is available, both restart cleanup and job restart can be accomplished online on a single screen.

1.3.8 Online Utility Execution

The CA-7 utilities allow online execution of commonly used IEHPROGM-type functions. Several utility commands are supported including Data Set Allocation, allocate and catalog, rename, scratch, uncatalog, list PDS and list catalog.

Online execution of utilities reduces time otherwise required to perform these functions, and eliminates dependence on other methods such as batch jobs, TSO, CA-Roscoe, and so forth.

1.3.9 Security

CA-7 supports both internal and external security. External security packages, such as CA-ACF2 and CA-Top Secret, can be used to control access to CA-7, which terminals an operator can use, which commands an operator can issue, which data sets an operator can access, and the authority associated to jobs submitted by CA-7. For detailed information about using external security with CA-7, refer to the *CA-7 Security Guide*.

An internal security matrix can be used if an external security package is not available or desired. With the CA-7 security matrix, five levels of security are provided:

- Terminal/Operator
- Operator/Application
- Application/Command
- Command/Function
- User ID/External Data Set

Through the security matrix, personnel are allowed to access only those functions of the system as defined by installation personnel. To further define security, there are exits available which allow interfacing with external security packages.

1.3.10 JCL Overrides

The JCL Override facility can dynamically include or omit override statements based on current execution requirements for a given job. Both scheduled and unscheduled override requirements are supported. CA-7 provides tools to validate the syntax of these statements.

Additionally, the CA-Driver component facilitates automatic manipulation of JCL during job submission. Some of the features include:

- Date manipulation in JCL (or in-stream data)
- Conditional expansion of JCL
- Using variable parameters in JCL
- Nesting procedures

1.3.11 External Communications Facilities

CA-7 provides several programs which facilitate communications between CA-7 and users outside the CA-7 address space (for example, batch jobs and online sessions). Among the programs provided for this purpose are SASSTRLR and U7SVC. These facilities flexibly accommodate a wide variety of needs but are most commonly used to communicate information to CA-7 about the availability of processing requirements for CA-7 jobs.

1.3.12 Batch Card Load Program

The Batch Card Load Program (BCLP) loads card-image data into data sets which may be specified as input requirements for CA-7 jobs. BCLP permits data set creation, replacement and modification by way of data set request cards. BCLP is one way that UCC7, the Central Control System, can be notified of the input requirements that permit it to coordinate the availability of input data with the jobs dependent on that data.

1.3.13 Workload Forecasting

The Workload Forecast facility provides several important functions. It allows you to:

- Project the scheduled workload for a given period of time.
- Verify newly defined schedules.
- Produce a hardcopy checklist of scheduled tasks in a worksheet format.

Criteria available for selecting data to be forecasted provide great flexibility in tailoring a forecast to meet specific needs.

1.3.14 Workload Planning

Workload planning is a modeling and simulation tool. Using an online forecast command, you can easily extract any subset of the workload or the entire workload definition. Once this workload definition has been created, it can be simulated with or without changes. Processing objectives and configurations can be handled in the same manner. Workload planning then simulates the production processing of the representative workload within the model processing objectives and configuration. The simulator can include work on request (that is, nonscheduled jobs) and rerun or restart work. Simulations are performed in batch mode. The CA-7 normal production control and scheduling functions are not affected by the planning function, so online processing continues as normal throughout the batch planning process.

1.3.15 Workload Balancing

Workload balancing dynamically balances CPU work based on user-defined processing objectives.

Through this facility, workload balancing and performance management objectives are predefined to CA-7. These objectives can be modified online at any time to accommodate situations which are not anticipated. CA-7 automatically selects and submits a mix of jobs to the host system which best meets the defined objective.

1.3.16 Workload Documentation

CA-7 allows documentation of the workload at many levels, from general descriptions to specific instructions. Documentation is added to the CA-7 workload definition through the CA-7 text editor and is available for reference at CA-7 terminals.

Including this documentation in the CA-7 database is optional. The documentation capability is provided to assist the production control staff. CA-7 does not require documentation to properly control work.

1.3.17 Management Level Reporting

CA-7 includes several tools that provide information on a data center's productivity. Management level reporting is done on a global basis for jobs and networks from historical data retained by CA-7. There are three categories of historical reports provided.

1.3.17.1 Automated Performance Analysis (APA) Reporting

APA will provide reports on performance information based on history. Information can be reported either online or in batch mode, and can range from the current date and time to two years prior. These reports are generated in summary graph format.

1.3.17.2 History Reporting

As work is processed under CA-7, activities are recorded in a Log data set. This information may be used to generate a variety of reports through the CA-7 History Reporting facility (for example, Work Scheduled and Not Run Report, Actual Processing Activity Report).

1.3.17.3 CA-Earl Reporting

An abbreviated yet powerful version of CA-Earl is provided with Unicenter TNG Framework for OS/390. With CA-Earl, several statistical type history analysis reports are provided. Some database inventory type reports are also provided.

1.3.17.4 CA-Easytrieve Plus Reporting

For clients with the CA-Easytrieve Plus product installed, report definitions are provided to produce the same statistical analysis and database inventory reports with CA-Easytrieve Plus as can be produced with CA-Earl.

1.3.18 CA-7 Text Editor

The CA-7 text editor is an interactive facility for creating, modifying and managing card-image data. With the text editor, job streams (JCL) can be created and submitted for processing online. It is also the mechanism for adding documentation (prose) to the CA-7 database. The ISPF editor replaces the CA-7 text editor if CA-7 is accessed through the TSO/ISPF interface.

1.4 Online Input

Prior to performing database maintenance, the user must be successfully logged on to a CA-7 terminal.

A series of formatted screens allow online input to the CA-7 database. Each major Database Maintenance (DB) command has an associated input screen. Input screens can be displayed by any of the following:

- Selecting from functions listed on DB Menu screens
- Entering the screen name as a top line command
- Entering the screen name in the FUNCTION field on any other menu or formatted input screen

When the desired command or menu screen option is entered, a data input screen or secondary menu is then displayed. For example, if function 4 is selected on the DB Menu, the DB.4 Workload Documentation Menu is then displayed.

The user may enter any CA-7 command on the top line of the current screen.

See DB Menu Screen on page 1-17 for a sample DB Menu screen.

After a DB function is processed from the formatted screen, the same screen is returned. This screen includes the function originally entered. If the same function is to be repeated, at least one character of that function must be reentered. This helps avoid inadvertent updates.

1.4.1 PF Keys

Once a function has been selected on the menu and the function screen is displayed, program function key 3, PF3 is temporarily set to return to the DB MENU screen. In native CA-7 VTAM mode, any value that was previously assigned to PF3, by either the user or CA-7, is temporarily ignored as long as the function screen is being used and reverts back to the original value after it is used once or after a top line command is entered.

With the TSO/ISPF interface, PF3 does not function in this way if any other value is assigned to PF3 when the user begins the session. The return-to-menu only works in the TSO/ISPF mode if PF3 remains unassigned in TSO/ISPF mode and is thus allowed to be returned back to CA-7 as a PF3.

PF7 and PF8 are similarly temporarily overridden to /PAGE-1 and /PAGE+1 respectively until PF3 is pressed or a top line command is issued.

1.4.2 Formatted Screen Messages

The following fields are common to all formatted input screens and appear at the bottom of each.

Field	Description
-------	-------------

PROGRAM:

Last 4 characters of the module name that processed this screen's data.

MSG-INDX: Return message identifier.

panel id Panel ID of this screen.

yy.ddd/hh:mm:ss

Date and time stamp of this returned message.

MSGs: Number of secondary output messages queued for this user's terminal. If no messages are queued, MSGS does not appear. This field also appears, immediately below the page number, on general inquiry displays whenever at least one message is awaiting display and the inquiry is made online. It does not appear on general inquiry output in batch mode.

MESSAGE: Up to two lines of message text. Messages are listed in the *CA-7 Message Guide* as a combination of the PROGRAM and MSG-INDX values. For example, SM11-00.

1.4.3 CA-7 Function Menu

To access the database maintenance functions, enter DB as the function on the CA-7 Function Menu and press Enter.

```
----- CA-7 FUNCTION MENU -----  
FUNCTION ==>  
  
  APA - AUTOMATED PERFORMANCE ANALYSIS  
  AR  - AUTOMATED RECOVERY FACILITY  
  DB  - DATABASE MAINTENANCE  
  QM  - QUEUE MAINTENANCE  
  RM  - VIRTUAL RESOURCE MANAGEMENT  
  UT  - UTILITIES  
  PS  - PERSONAL SCHEDULING  
  
  HELP - TUTORIAL  
  
  
PROGRAM: MNU0  MSG-INDX: 00  -- MENU      --  yy.ddd / hh:mm:ss  
MESSAGE: SPECIFY DESIRED FUNCTION OR ENTER A COMMAND ON THE TOP LINE
```

1.4.4 DB Menu Screen

Use the DB Menu screen to select various database maintenance functions.

```

----- CA-7 DATABASE MAINTENANCE MENU -----
FUNCTION ==>

DATA BASE DEFINITION FOR:
 1 - CPU JOB
 2 - SCHEDULING
 3 - JOB PREDECESSOR/SUCCESSOR
 4 - WORKLOAD DOCUMENTATION
 5 - INPUT/OUTPUT NETWORK
 6 - DATA SET

OTHER FUNCTIONS AVAILABLE:
 7 - JCL LIBRARY MAINTENANCE
 8 - TEXT EDITOR
 9 - CLEAR THE TEXT EDITOR ACTIVE AREA
    ACTIVE AREA NOW CONTAINS 0000 LINES OF TEXT

PROGRAM: SDMO MSG-INDX: nn  -- DB      -- yy.ddd / hh:mm:ss
MESSAGE:

```

To display, enter:

- DBM (or just DB) as a top line command.
- DBM (or just DB) as the FUNCTION value on any other menu or formatted input screen.

To exit, enter:

- The name of an online screen as the FUNCTION value or move the cursor to the top line and enter a top line command if some other function is desired.

1.4.4.1 Usage Notes

The desired function is selected by entering its value as the FUNCTION and pressing the Enter key.

1.4.5 Bypassing the Menu

CA-7 provides extensive menu service for Database Maintenance. The menus are structured to help a novice user step through various screens. However, as proficiency increases, it may be desirable to bypass primary menus and proceed directly to the desired screen.

The primary DB screen may be bypassed by entering a command screen name on the top line of the current screen or as the FUNCTION value on any other menu or formatted screen. For example, DB.1, DB.4, and so forth.

1.4.5.1 Bypassing Secondary Menus

Functions 2, 3, and 4 have secondary menu screens. To go directly from the DB Menu screen to a function that appears on one of the secondary menus, enter FUNCTION in n.n format. For example:

```
-----
FUNCTION ==> 2.3
```

on the DB Menu screen indicates that you want function 2 (SCHEDULING) and that Scheduling Menu function 3 (OUTPUT NETWORK) is the function that is to be performed. CA-7 displays the DB.2.3 screen directly and bypasses the Scheduling Menu (DB.2).

You may also bypass these secondary menu screens by entering the equivalent top line command, placing a comma after the command and selecting a category, as follows:

- SCHD may be followed by JOB, INWK, ONWK, JTRG, DTRG, or NTRG.
- JOBCONN may be followed by DSN, JOB, NWK, RPT, or USER.
- PROSE may be followed by JOB, DSN, SYS, USER, NWK, or DD.

For example, the top line command JOBCONN,JOB would display the DB.3.2 screen for defining, listing, or updating job predecessors without going through the DB or Job Predecessor/Successor Menu screens.

1.4.6 Function Shortcuts

An online shortcut for requesting LIST functions of job predecessor/successors and schedule triggers is also available. In this case, keyword values identifying the element data desired are included in the top line command following the category, as follows:

- JOBCONN,DSN may be followed by JOB=jobname and SCHID=nnn.

For example:

```
JOBCONN,DSN,JOB=BACKUP,SCHID=2
```

lists all DSN connections to job BACKUP for schedule ID 2 with no further input needed.

- The SCHD trigger function may be followed by JOB=jobname, DSN=dataset-name/DSNBR=dataset-number or NWK=network-name/DSNBR=network-number (depending on the trigger function).

For example:

```
SCHD,NTRG,NWK=PAYPUNCH
```

lists all jobs and schedules triggered by the completion of input network PAYPUNCH.

1.4.6.1 Shortcut Examples

The following are additional shortcut examples:

```
JCL,member,JCLID=nnn
```

Entering this on the top line (where JCLID defaults to zero) has the same result as the following three steps:

1. DB.7 top line command
2. FETCH function with member and JCLID of nnn
3. EDIT function

```
JOB,jobname
```

Entering this on the top line has the same result as the following two steps:

1. DB.1 top line command
2. LIST function with job name

1.4.7 Function Transfer

To transfer from one menu or formatted screen to another, enter the screen name in the FUNCTION field of the current screen. For example, you may transfer to Queue Maintenance by entering QM in the FUNCTION field of the screen.

1.4.8 Aliases for Formatted Screen Functions

All function values for formatted screens may be assigned alternate (alias) names. This allows for user-designated abbreviations (for example, L for LIST). It also permits alternate values such as CHANGE for UPD. The only restrictions are that the alias names must be 8 characters or less, and must not conflict with a function or other alias names. The following is a list of formatted screen function values and some alias names which are distributed with CA-7:

Function(s)	Service level	Alias
ADD	ADD	A,ADDT,AELETE,AIST,APD
APPEND	READ	AP,APP
APPENDP	READ	n/a
CLEAR	n/a	CL,CLR
DD	DELETE	n/a
DELETE	DELETE	D,DEL,DELT
DELPRRN	UPDATE	n/a
EDIT	n/a	E,EDITH
EXIT	n/a	n/a
FE	READ	FEIT,FEPL,FEVE
FETCH	READ	F
FETCHP	READ	FP
FORMAT	n/a	FMT,FOR,FORM
FPE	READ	n/a
FREE	DELETE	n/a
LIST	READ	L,LDD,LDIT,LISTA,LISTP,LISTR,LPD
RENAME	UPDATE	REN
REPL	UPDATE	R,REP
REQ	UPDATE	n/a
RESOLV	SUBMIT	RES
RET	SUBMIT	n/a
RUN	SUBMIT	n/a
RUNH	SUBMIT	n/a
SAVE	ADD	S
SR	UPDATE	n/a
SS	ADD	n/a
SUBMIT	SUBMIT	SUB

Function(s)	Service level	Alias
UPD	UPDATE	U,UDD,UIST,UPDATE,UPDT
XPOST	UPDATE	n/a
XPRE	UPDATE	n/a
XQ	UPDATE	n/a
XQJ	UPDATE	n/a
XQM	UPDATE	n/a
XQN	UPDATE	n/a
XRQ	UPDATE	n/a
XRST	UPDATE	n/a
XSPOST	UPDATE	n/a
XSPRE	UPDATE	n/a
XUPD	UPDATE	n/a
XWLP	UPDATE	n/a

Steps for modifying the alias values are given in the *CA-7 Systems Programmer Guide*, Chapter 9.

1.5 Batch Input

CA-7 supports two different formats for transaction input and output.

The ONLINE format is used in CA-7 online terminal sessions. Online terminal sessions support command line input and output as well as data transfer using formatted screens.

Unless the online terminal session is in text editor mode, all input beginning in the upper left corner of the screen is interpreted as command line input. Transactions entered in this area are referred to here as command transactions.

A command transaction begins with the command name and may be followed by parameters. These transactions are documented in the *CA-7 Commands Guide*.

Formatted screen transactions differ from command transactions. Input for formatted screen functions is not solicited at the command line. Formatted screens structure input and output in an online terminal session using delimited and tagged screen fields. Certain functions require formatted screen transactions. Most database maintenance functions require transactions in this mode.

The BATCH format is used by CA-7 external communicators such as the Batch Terminal Interface (BTI) and the CCI Terminal Interface.

Command transactions are entered in batch format just as they would be in an online terminal session. However, functions that use formatted screens in the online environment have special syntax requirements in batch, because formatted screens cannot be displayed in batch.

In those cases where batch and online transaction formats differ, the description of the corresponding batch transaction format is included in the documentation of each online function.

See the *CA-7 Interfaces Guide* for further information on external communicators.

1.5.1 Function Shortcuts and Bypassing the Menu

The shortcuts available online to bypass certain screens are of limited value in batch. Batch transactions should single step from function to function much as a new user would step through a sequence of equivalent screens (for example, DBM, SCHD, JOB, and so forth). Some commands which provide secondary menu screens online, such as DB.2, DB.3 and DB.4, use positional keywords to identify the specific function rather than requiring another command to be entered separately.

1.5.2 Function Transfer

To transfer to another DB function, input a record with the corresponding batch screen name starting in the first column. See the N220 installation job for an example of function transfer between DB mode and top line command mode. To exit from Database Maintenance using batch input, the user must input a record with DBM starting in the first column.

1.5.3 Aliases for Formatted Screen Functions

Batch commands may use any of the alias names that were assigned just as they are used online. For a list of alias names supplied with CA-7, see 1.4.8, “Aliases for Formatted Screen Functions” on page 1-20.

See the "User Exits and Modifications" chapter of the *CA-7 Systems Programmer Guide* for the procedures for assigning alias names.

Chapter 2. Jobs

This chapter describes setting up and maintaining a job in the CA-7 database.

2.1 Adding a Job

You can add jobs in several ways. The primary method is entering data on the DB.1 screen. However, there are other methods.

If the job information already exists in machine-readable form, a program can be written to perform the DB.1 screen functions using batch commands with a Batch Terminal Interface (BTI) job.

For further details on BTI, see the *CA-7 Interfaces Guide* and 1.5, “Batch Input” on page 1-22.

If the JCL library has been defined to CA-7 but the jobs have not been added to the database, the user may use the DEMAND command to add the job to the database. This command causes the job to be executed. The execution includes a LOAD step since the job is not defined in the database. The LOAD command can also be used to submit the jobs but they do not execute. When scheduled using the LOAD command, the LOAD step executes but the rest of the job's JCL is flushed.

To use the DEMAND or LOAD commands to add the job, certain conditions have to be met.

1. The job name of the job to be added must be the same as the JCL member name in the JCL library.
2. Job characteristics must be satisfied by the default values defined for the DB.1 screen. The CA-7 default values for the DB.1 screen can be overridden by defining a job named DEFAULTS with the values wanted. System defaults cannot be overridden with the DEMAND and LOAD commands. Job characteristics, not covered by the defaults, must be manually checked.
3. The JCL library ID number or symbolic JCL library name must be specified on the command. One JCL library should be given an index value of 0 (zero) so that the DB.1 screen JCLID field does not have to be entered for jobs with JCL residing in that library when issuing the top line JCL command.

The DEMAND command represents on-request scheduling which is useful on a first-time basis. As an ongoing practice, it is not recommended except for jobs which must always be handled as on-request work.

2.2 DB.1 - CPU Job Definition Screen

Use the DB.1 screen for entering or reviewing data related to CPU jobs.

```

----- CA-7 CPU JOB DEFINITION -----
FUNCTION: xxxxxxxx (ADD,DELETE,DD,PURGE,DELPUR,FORMAT,LIST,UPD)
JOB: xxxxxxxx
GENERAL:      SYSTEM: xxxxxxxx  JOBNET: xxxxxxxx  OWNER: xxxxxxxx  UID: nnn

JCL:          ID: nnn  MEMBER: xxxxxxxx  RELOAD: x  EXEC: x  RETAIN-JCL: x
              LIB: xxxxxxxxxxxxxxxx

REQUIREMENTS: HOLD: x  JCL-OVRD: x  USE-OVRD-LIB: x  VERIFY: x  MAINT: x
                SATISFACTION LEAD-TIME: JOB: nn  DSN: nn  ARFSET: xxxxxxxx

EXECUTION:     MAINID: xxxx  INSERT-RMS: x  COND-CODE: nnnn  RO: xx
                DONT SCHEDULE -- BEFORE: yyddd hhmm  AFTER: yyddd hhmm

MESSAGES:      LTERM: xxxxxxxx  REQUIREMENT-LIST: x  PROMPTS: x
                ERROR MSGS -- RQMTS NOT USED: x  DSN NOT FOUND: x

RESOURCES:     REGION: nnnn  CLOCK-TIME: hhmm  CPU-TIME: mmmss
                CLASS: x  PRTY: nnn  MSGCLASS: x
                TAPE DRIVES...TYPE1: nnn M nnn C  TYPE2: nnn M nnn C

PROGRAM: SM20  MSG-INDX: nn  -- DB.1  -- yy.ddd / hh:mm:ss
MESSAGE:

```

To display, enter:

- 1 as the FUNCTION on the Database Maintenance (DBM) Menu screen.
- DB.1 as the FUNCTION value on any other menu or formatted input screen.
- DB.1 or JOB as a top line command.

To exit:

- Press the PF3 key to return to the DBM Menu screen except with the TSO-ISPF interface when the PF3 key has been assigned to some other function.
- Move the cursor to the top line of the screen and enter the desired top line command if yet another function is desired.

2.2.1 Field Descriptions

FUNCTION:

The function to be performed. Value must be the name of some other screen or one of the following:

ADD	Adds a job to the database.
DELETE	Deletes a job and its associated data from the database (documentation, schedules, and so forth).
DD	Same as DELETE, but deletes any data sets which this job references if the data sets have no users and no documentation, and do not cause job triggering. This includes all virtual resources connected to the job.
PURGE	Same as DD, but also deletes job trigger definitions that trigger the job being deleted, job requirement definitions that require the job being deleted; and, if the CA-11 interface is active, deletes the CA-11 CMT member for the job being deleted.
DELPRRN	Deletes the JCL saved in the trailer queue from the job's prior run.
FORMAT	Clears the screen of all input data.
LIST	Lists a job and job related information. In batch, a formatted screen is not listed; only a found or not found message is returned.
UPD	Updates database information about a job.

Required: Yes

Batch keyword: Positional parameter

Note: A default interpretation can be set for the DELETE function in the CA-7 initialization file. See the OPTIONS statement in Chapter 5, "Initialization," of the *CA-7 Systems Programmer Guide*.

Also, when using the PURGE function, some residual elements may not be deleted if the user issuing the PURGE does not have update access to the other jobs affected. See the *CA-7 Security Guide*, Chapter 3, "CA-7 Security Initialization Options," BYPSEC= keyword for information on bypassing these security checks.

JOB:	<p>The job name on which the indicated function is to be performed.</p> <p>Size/Type: 1 to 8 alphanumeric characters</p> <p>Required: Yes</p> <p>Batch keyword: Positional parameter</p> <p>Note: Job name format of UCC7Rxxx (where xxx is any 3 characters) is reserved for use with Workload Balancing. This imposes a restriction that no other user-defined job can begin with UCC7R as the first 5 characters.</p> <p>This job name always overlays the job name on the JOB statement in the execution JCL.</p>
GENERAL:	Indicates that this section of the screen contains general information about the job. No input is allowed for this field.
SYSTEM:	<p>The user-defined application system name of which this job is a part. This field cannot contain a comma.</p> <p>Size/Type: 1 to 8 alphanumeric characters</p> <p>Required: No</p> <p>Batch keyword: SYSTEM</p>
JOBNET:	<p>The name of a CPU job network of which this job is a part. This field cannot contain a comma.</p> <p>Size/Type: 1 to 8 alphanumeric characters</p> <p>Required: No</p> <p>Batch keyword: JOBNET</p>
OWNER:	<p>ID identifying ownership of this job. Depending on the CA-7 startup options taken, the external security product being used and contents of the JCL, this value may be offered to the external security package by CA-7 at job submission time as the authority for this job to be executed. Although this field supports up to 8 characters, some external security packages only accept up to 7 characters. This field must not exceed any such limit that exists.</p> <p>Size/Type: 1 to 8 alphanumeric characters</p> <p>Required: No</p> <p>Batch keyword: OWNER</p>
UID:	<p>The CA-7 user security identification.</p> <p>Size/Type: 1 to 3 numeric characters from 0 to 255</p> <p>Default: 0 (No internal security protection)</p> <p>Required: No</p> <p>Batch keyword: USERID</p>
JCL:	Indicates that this line of the screen contains JCL information about the job. No input is allowed for this field.

ID:	<p>A numeric index value associated with a JCL library. Two values have special significance: 254 is reserved to indicate the override library (see USE-OVRD-LIB field for additional information) and 255 is reserved for use with the HELP library. 255 is also assigned to libraries that are referenced using a symbolic index (see LIB field for additional information). LIB and ID are mutually exclusive.</p> <p>Size/Type: 1 to 3 numeric characters from 0 to 253 Default: 0 Required: No Batch keyword: JCLID</p>
MEMBER:	<p>The JCL library member name.</p> <p>Size/Type: 1 to 8 alphanumeric characters Default: Job name Required: No (required if member name and job name are different) Batch keyword: JCLMBR</p>
RELOAD:	<p>Indicates if this job's JCL is to be reloaded (Y, N, or X). When a job comes into the request queue, it is either flagged for load processing or not flagged. If RELOAD=X, then the job is not flagged for load processing unless the LOAD command is used. If RELOAD=Y, then the job is flagged for load processing. If RELOAD=N, then the job is not flagged for load processing unless it is the first time it has run in CA-7. A RELOAD=Y is automatically reset to N once the load completes successfully. A RELOAD=X is never automatically changed even if the LOAD command is used.</p> <p>Default: N Required: No Batch keyword: RELOAD</p>
EXEC:	<p>Indicates whether this job should be executed (Y or N). If N is used, the job does not run but shows a normal completion as if it did run. JCL is not required for nonexecutable jobs. VRM definitions for the job are ignored if EXEC of N is used.</p> <p>Note: ARF recovery will not be invoked for non-executable jobs.</p> <p>Default: Y Required: No Batch keyword: EXEC</p>
RETAIN-JCL:	<p>Indicates whether execution JCL is to be retained in the trailer queue after a successful run (Y or N).</p> <p>Default: N Required: No Batch keyword: RETJCL</p>

LIB: JCL library identification. Must be a symbolic INDEX assigned to a JCL statement. Symbolic value &HELP is reserved for the HELP library. LIB and ID are mutually exclusive.

Size/Type: 2 to 16 alphanumeric characters beginning with
ampersand (&)

Required: No

Batch keyword: JCLLIB

Note: Schedule scan uses the current specification for LIB when attempting to attach the JCL for a job in the request queue with RETRY status.

REQUIREMENTS:

Indicates that this section of the screen contains requirements information about the job. No input is allowed for this field.

HOLD: Indicates whether this job is to be placed in a hold status when it enters the request queue (Y or N).

Default: N

Required: No

Batch keyword: HOLD

JCL-OVRD: Indicates whether this job needs manual JCL overrides before it can be submitted (Y or N). Similar to the JCLOVRD command.

Default: N

Required: No

Batch keyword: JCLOVR

USE-OVRD-LIB: Indicates whether the JCL is to be retrieved from the JCL Override library (JCLID=254) for the next run only (Y or N). This field is automatically set back to N the next time the job comes into the request queue.

Default: N

Required: No

Batch keyword: JOVRDLIB

VERIFY: Indicates whether this job requires any presubmission manual verification (Y or N). Similar to VERIFY command.

Default: N

Required: No

Batch keyword: VERIFY

MAINT:

Indicates whether this job is a maintenance job (for example, a system utility) with no production data set requirements (Y or N). If MAINT=Y, then all input data set requirements are ignored. None of the output data sets created by this job are posted back to CA-7.

Marking a job as MAINT allows job triggering but not data set triggering.

In addition, if the LOADDNS keyword is used on the DBASE statement in the initialization file, then the LOAD process does not build any DD or data set information for jobs that are marked MAINT=Y on the DB.1 (JOB) screen. This means that there will not be any data set connections for these jobs unless added manually.

Default: N
 Required: No
 Batch keyword: MAINT

SATISFACTION LEAD-TIME:

Indicates that this area of the screen contains lead time information about the job requirements. No input is allowed for this field.

JOB:

The number of hours to be considered when satisfying job dependent requirements. Values are:

Default: 0
 Batch keyword: JOBLDTM

0 Indicates no lead time is to be considered when satisfying this job's requirements.

99 The requirement is never to be considered as already satisfied when the job enters the queues. Each predecessor job must complete normally while this job is in the request queue.

nn Since the last run of this job, each predecessor job must have run within the last nn hours. Values for nn may be from 1 to 98.

Note: JOB and DSN are on a global level for all job and data set requirements which have a LEADTM value of 0000 on the DB.3 screens. This applies to all SCHIDs that the job runs under. See 4.3.1, "DB.3 - Job Predecessor/Successor Menu Screen" on page 4-6.

DSN:

The number of hours to be considered when satisfying data set requirements. For a description of values, see preceding field name (JOB).

Size/Type: 1 to 2 numeric characters from 0-99
 Default: 0 (indicates ignore this field)
 Required: No
 Batch keyword: DSNLDTM

- ARFSET:** Names the collection of ARF definitions that apply to this job. Remember that ARF recovery will not be invoked for non-executable jobs.
- Size/Type: 1 to 8 alphanumeric characters
 Required: No
 Batch keyword: ARFSET
- Note:** The specified ARFSET must be defined in the ARF database.
- EXECUTION:** Indicates this screen section contains execution information about the job. No input is allowed for this field.
- MAINID:** Indicates on which CPU the job may or may not be scheduled. If the job requirements impose CPU restrictions, specify SYn or /SYn where n is the system number and / indicates not this system. System numbers must be consistent with the initialization file CPU statement MAINIDS parameters.
- Size/Type: 1 to 4 alphanumeric characters
 Default: ALL (allows job to run on any CPU)
 Required: No
 Batch keyword: MAINID
- Note:** If a triggered job does not specify a MAINID, the job runs with the MAINID specified by the triggering job.
- INSERT-RMS:** Indicates if the CA-11 RMS step is to be automatically inserted at execution time by CA-7. Specify Y to insert the step with the CA-11 processing code of P. Enter N to indicate the RMS step is not to be inserted. Unless a different name is specified with the PROCRMS parameter on the RESTART statement of the initialization file, the default procedure name defined in the CA-11 Option Table is inserted. See the *CA-7 Interfaces Guide* for more information.
- Default: N
 Required: No
 Batch keyword: INSRTRMS or RESTART
- COND-CODE:** Used with RO (relational operator) to define the job-level condition codes used to determine if a job executes successfully or not.
- Size/Type: 1 to 4 numeric characters from 0-4095
 Default: 0
 Required: No
 Batch keyword: CONDCODE

RO: Indicates the relational operator of the condition code (COND-CODE) or if the step level #SCC statements are being used in the job's JCL. Values are:

EQ Equal to

LT Less than

GT Greater than

GE Greater than or equal to

LE Less than or equal to

NE Not equal to

#S Step condition code tests to be made based on #SCC statements in the JCL. See 10.3.4, “#SCC Statement” on page 10-18 for the use of the #SCC statement.

IG No evaluation of the job is done. CA-7 always assumes the job completes successfully, regardless of condition codes,abend codes, or run-time JCL errors. When this is used, the INSERT-RMS fields should be N.

0 No condition test is to be made.

Default: 0

Required: No

Batch keyword: RELOPR

If 0 is used, no test is made on the job's condition code.

The highest condition code that this job generates is tested by this pair of parameters. For example, if COND-CODE is set at 8 and RO is set at LT, the job is marked as completing abnormally if 8 is less than the job's highest condition code.

Note: This test is for CA-7's internal use only. It simply tells CA-7 what action to take after the job completes. CA-7 does not determine or control which steps are to be executed.

DONT SCHEDULE -- BEFORE:

Indicates that this job is not to be scheduled before this date and time. Especially helpful for timing the start of new jobs. Screen input is:

BEFORE: yyddd hhmm

Where yyddd is the Julian date and hhmm is the time of day. Batch input is:

BDATE=yyddd,BTIME=hhmm

Size/Type: Julian date specified as yyddd and time specified as hhmm

Default: All zeros

Required: No

Batch keyword: BDATE and BTIME

AFTER:

Indicates that this job is not to be scheduled after this date and time. Especially helpful for timing permanent discontinuation of a job. Screen input is:

AFTER: yyddd hhmm

Where yyddd is the Julian date and hhmm is the time of day. Batch input is:

ADATE=yyddd,ATIME=hhmm

Size/Type: Julian date specified as yyddd and time specified as hhmm

Default: 99999 0000

Required: No

Batch keyword: ADATE and ATIME

Note: If BDATE and BTIME values are equal to ADATE and ATIME, the job is not scheduled and does not appear on forecasts. If BDATE and BTIME values are greater than ADATE and ATIME, the job is not available for scheduling during the inclusive times only. Thus, if BDATE=00031 and BTIME=0000 and ADATE=00001 and ATIME=0000, the job does not schedule during the time from Jan. 1, 2000 through Jan. 31, 2000, but does schedule at all other times.

CA-7 uses current date and time for comparisons.

MESSAGES:

Indicates that these lines of the screen contain information about messages which may occur for the job. No input is allowed for this field.

LTERM: Messages about this job are to be routed to this logical terminal name.

Size/Type: 1 to 8 alphanumeric characters

Default: If not entered, the LTERM associated with the JCL library in the initialization file JCL statement is used. If LTERM is not specified on the JCL statement, the default is MASTER.

Required: No

Batch keyword: LTERM

REQUIREMENT - LIST:

Indicates whether preexecution requirements are to be listed for this job when it enters the request queue (Y or N).

Default: Y

Required: No

Batch keyword: RQLST

PROMPTS:

Indicates if prompt messages are to be issued if this job is late (Y or N).

Default: Y

Required: No

Batch keyword: PROMPT

Note: If LATEPROMPT=LATE is specified on the OPTIONS statement of the CA-7 initialization file, setting this value to no (N) will prevent the job from ever having a status of LATE on an LQ or LRLOG display. Jobs defined with a value of yes (Y) will be processed the same regardless of the LATEPROMPT setting.

ERROR MSGS -- RQMTS NOT USED:

Indicates whether error messages for job requirements not used are to be issued (Y or N).

Default: Y

Required: No

Batch keyword: RQMSG

DSN NOT FOUND:

Indicates whether error messages for data sets used at execution time but not found in the CA-7 database are to be listed (Y or N).

Default: Y (The messages are not issued if PERFORM=1 is specified on the INIT statement in the initialization file.)

Required: No

Batch keyword: DSNMSG

RESOURCES:

These lines contain resource information about the job. No input is allowed in this field.

- REGION:** The region size required by this job. For information only.
- Size/Type: 1 to 4 numeric characters
 Default: 0
 Required: No
 Batch keyword: REGION
- CLOCK-TIME:** CA-7 maintains certain SMF feedback data in its database, including a weighted average elapsed runtime. If the database is updated with a time of 0000, the counters for number of runs, number of times late, and number of restarts are also reset to 0000. One use of this value is deadline prompting. If 2359 is specified, deadline prompt time is not adjusted. It remains due-out time minus lead time.
- Note:** Clock time and CPU time averages are not updated if the job either fails (abend, JCL error, and so on) or is restarted. The weighted average is the value in the database times 5, plus the value from the current run, divided by 6. This tends to keep the current run from possibly skewing the value.
- Size/Type: 4 numeric characters specified as hhmm, where hh can be 00 through 23 and mm can be from 00 to 59
 Default: 0000
 Required: No
 Batch keyword: ELAPTM
- CPU-TIME:** CA-7 maintains weighted average CPU time usage values for each job in its database. (See the preceding note under CLOCK-TIME.)
- Size/Type: 5 numeric characters specified as mmmss, where mmm is minutes and ss is seconds
 Batch keyword: CPUTM
- CLASS:** Indicates the CA-7 WLB job class. If using Workload Balancing, any job without a specified job class is automatically assigned to class A. Also, for workload balancing, any job that enters the queue as a result of a RUN(H) command is assigned to class 9. Any job that enters the queue as a result of a LOAD(H) command is assigned to class 8. Does not have to match the JOB statement CLASS value.
- Size/Type: 1 alphanumeric character
 Default: A space
 Required: No
 Batch keyword: CLASS

PRTY:	Indicates the CA-7 WLB job priority. A value of 255 indicates an "express" priority used to bypass WLB priority scheduling criteria. If using WLB, any job without a specified priority is automatically assigned a priority of 100 unless the default is changed. Size/Type: 1 to 3 numeric characters from 0 to 255 Default: 0 Required: No Batch keyword: PRTY
MSGCLASS:	Indicates the job's message class. Informational only. Even though this field can be changed, it will not cause the JCL to be changed. Also, no validity checking is done on this field. Size/Type: 1 alphanumeric character Default: A space Required: No Batch keyword: MSGCLASS
TAPE DRIVES...	Indicates that this line of the screen contains information about tape drives needed for the job. (If not using Workload Balancing, then these fields are informational only.) No input is allowed for this field.
TYPE1:	Indicates that the two following fields (M and C) contain information about TYPE1 tape drives needed for the job. No input is allowed for this field.
M	Indicates a manual override value for the number of TYPE1 tape drives needed for the job. Normally this field is only used to update a job where tape drive requirements have been significantly changed, higher or lower, and the job has not been reloaded since the change. A value of 255 can be used to specify that the job uses 0 TYPE1 tape drives. Size/Type: 1 to 3 numeric characters from 0 to 255 Default: 0 Required: No Batch keyword: TP1M
C	This field indicates the calculated value for the number of TYPE1 tape drives needed for the job. Value is automatically calculated when the job is loaded (or reloaded) or the RESANL command is performed on the job. Calculation is based on DD references whose unit values are defined as TYPE1 tape drives in the SASSUTBL module. This field is display only and no input is allowed.
TYPE2:	Indicates that the following two fields (M and C) contain information about TYPE2 tape drives needed for the job. No input is allowed for this field.

M	Indicates a manual override value for the number of TYPE2 tape drives needed for the job. Normally this field is only used to update a job where tape drive requirements have been significantly changed, higher or lower, and the job has not been reloaded since the change. A value of 255 can be used to specify that the job uses 0 TYPE2 tape drives.
	Size/Type: 1 to 3 numeric characters from 0 to 255
	Default: 0
	Required: No
	Batch keyword: TP2M
C	This field indicates the calculated value for the number of TYPE2 tape drives needed for the job. Value is automatically calculated when the job is loaded (or reloaded) or the RESANL command is performed on the job. Calculation is done as described in TYPE1. This field is display only and no input is allowed.

2.2.2 Usage Notes

Many of the fields have predefined default values which are assigned if no value is entered. Once a job has been added, the LIST function returns this screen with all fields containing either the specified data or the defaults taken.

If different default values are desired, a job with the desired default values can be added and given the name DEFAULTS. All future adds, either online or batch, then use that job for default values. If some other job name is desired for the defaults pattern job, the initialization file DEFAULTJOB parameter must be set to the correct job name.

You cannot set the UID value with a DEFAULTS job.

If you wish to control which jobs are eligible to be marked as LATE, you can use the LATEPROMPT= keyword on the OPTIONS statement in the CA-7 initialization file. If you specify LATEPROMPT=LATE, then jobs which are defined with 'PROMPT: N' will never show a status of LATE on LQ displays (including LREQ, LRDY, LACT, and so on). This allows your operations staff to concentrate on important jobs when they show up as LATE.

2.2.3 Batch Input Example

This batch input example adds job CA07XX01 specifying a system, a JCLID and a class:

```
JOB
ADD,CA07XX01,SYSTEM=TESTNTWK,JCLID=3,CLASS=A
DBM
```

See the previous discussions of these screen fields for details on each.

2.3 Changing a Job

The basic way to change a job's characteristics, which appear on the DB.1 screen, is to use the UPD function. However, you can make wholesale changes to many jobs using the Batch Terminal Interface (BTI) facility. For further details on BTI, see the *CA-7 Interfaces Guide* and 1.5, “Batch Input” on page 1-22. Refer also to the Database Transportability chapter of this guide for a discussion of another tool that can be very helpful when making extensive changes.

The database information should always agree with the JCL for the job. Therefore, whenever permanent changes are made to the JCL, you should resynchronize the database with new JCL. The CA-7 LOAD process accomplishes this, and you can handle it in one of several ways.

- Manually issue a LOAD command after the JCL has been changed.
- Set the RELOAD value on the DB.1 screen to Y if the change is made outside of CA-7 (for example, through TSO).
- Make any permanent changes to the JCL through the DB.7 screen. This causes the RELOAD field on the DB.1 screen to be set to Y automatically.

Note: The DB.3 screens establish dependency relationships for CPU jobs, workstation networks, and so forth, thereby defining prerequisites. Whenever the JCL changes for any job, the user must consider whether those changes affect the dependency relationships for the job. See 4.3.1, “DB.3 - Job Predecessor/Successor Menu Screen” on page 4-6.

Chapter 3. Scheduling

This chapter identifies the screens and commands used to schedule work in CA-7.

3.1 Work Scheduling

CA-7 uses both date/time-driven and event-driven facilities to cause work to be scheduled and initiated. To CA-7, work is defined as CPU jobs and/or preprocessing and postprocessing workstation networks. Event-driven scheduling, known as trigger scheduling, is a far more efficient technique than date/time oriented schedules. See 3.1.3, “Event Scheduling” on page 3-6 for more details on this technique.

For CPU jobs and preprocessing networks, date/time-driven schedules are generated through definition of base calendars and definition of schedules relative to base calendars. Once defined, work is automatically scheduled based on defined schedules and current date and time values within the CA-7 system. For postprocessing networks, output workstations are scheduled based on the scheduling of the CPU job to which the networks are connected.

Event-driven scheduling causes jobs to be scheduled or initiated based on completion of a previous activity or event. Event mechanisms are the creation of required data sets, completion of predecessor jobs, or completion of a preprocessing network.

Time-driven and event-driven facilities may be used in combination when scheduling and initiating work. For example, a CPU job may have a timed schedule for processing but may also have dependencies on data sets created by another job. When combined, time and event requirements must be satisfied before work is initiated.

The user should be aware that the schedule ID offers a powerful scheduling tool. Requirements, networks, JCL overrides, job triggers, and so forth, can all be varied through the use of different schedule IDs. For example, if a given job is to run on Monday through Friday but the requirements vary on Fridays, this same job could have a different schedule ID defined for its Friday run.

Multiple schedule options may be chosen for a single schedule ID (for example, daily and monthly). However, a job which is scheduled automatically on a defined date/time basis is not scheduled more than once per day for a given schedule ID.

3.1.1 Date and Time Scheduling

Date/time-driven scheduling is performed by the schedule scan function of CA-7 based on initialization file parameters and database definitions. By combining base calendar and schedule definitions through a resolution process, CA-7 stores information about date/time related work in its database.

3.1.1.1 Base Calendars

Base calendars define available processing days for the installation. Any number of base calendars may be defined. However, it may be possible to satisfy installation requirements with only one or two calendars.

Each calendar defines a single year comprised of twelve months with a starting and year-ending boundary that can be crossed only in the first or twelfth month. A given calendar may represent schedule months in either standard or user-defined format (for example, accounting, fiscal, and so forth).

Base calendars can be generated through either batch or online facilities. To use the online facility (DB.2.8), a CA-7 calendar PDS must be allocated and identified to CA-7. See the *CA-7 Systems Programmer Guide* for information on enabling online calendar maintenance. Processing of base calendars for resolution of schedules is the same regardless of whether they were generated using the batch or online facilities.

Calendars can be generated through the CA-7 CALENDAR macro. Macro keywords are used for calendar definition and the macro is assembled and link edited using standard procedures. Base calendars are link edited to the CA-7 LOADLIB or a specified calendar library, and are stored in the form of bit masks which represent available and unavailable processing days for the year.

Timing for generation of base calendars is flexible. The only requirement is that a particular calendar exist before resolution of work schedules related to the calendar is attempted. Base calendars must be regenerated whenever available processing days change and before the beginning of each year. Generally, next year's base calendar(s) should be generated by July 1 of the current year. New base calendars may be generated at any time without impacting current processing schedules. Schedule resolution, described later, is used to change the processing schedules, for jobs or networks, with a new year or changed base calendars.

PRINT Command:

Use the PRINT command to produce a month-by-month listing of indicated base calendars. PRINT produces output which reflects the beginning and ending days of each month, holidays, or any other nonprocessing days that were defined when the calendar was produced. The PRINT command is described in the *CA-7 Commands Guide*.

3.1.1.2 Schedule Definition

Definition of work schedules is accomplished through DB.2 screens. This process involves identifying when work is to be processed in relation to given calendars. When a schedule is initially defined (or changed), schedule resolution with the base calendar(s) is used to complete the definition of the processing schedule.

3.1.1.3 Schedule Resolution

Once calendars and schedules are defined, schedules are resolved against calendars to ensure compatibility and create the processing schedule. This is accomplished through the RESOLV facility which is available as both a top line command and a function on the DB.2 screens. The result of this resolution is the schedule used by CA-7 to schedule preprocessing work and jobs. Figure 3-1 on page 3-5 illustrates this function. The figure demonstrates the path for a base calendar generated through the batch facility. Calendars generated through the online base calendar maintenance facility (DB.2.8) take a different path leading up to the *resolution* phase but are treated the same as batch calendars in the actual resolution of schedules.

RESOLV Command:

Use the top line RESOLV command to create or modify processing schedules for jobs or workstation networks which are to be scheduled on a date/time basis. Work which is scheduled by a trigger or on-request work which is DEMANDED or RUN has no direct relationship to a base calendar and therefore does not require the use of this function.

Whenever a calendar-oriented processing schedule expires (at the end of June or the end of the year) or the scheduling frequency is redefined for any other reason, it again becomes necessary to resolve the schedule. Options on the RESOLV command are provided for resolving all or any subset of those tasks scheduled with a single command. The RESOLV command is fully described in the *CA-7 Commands Guide*.

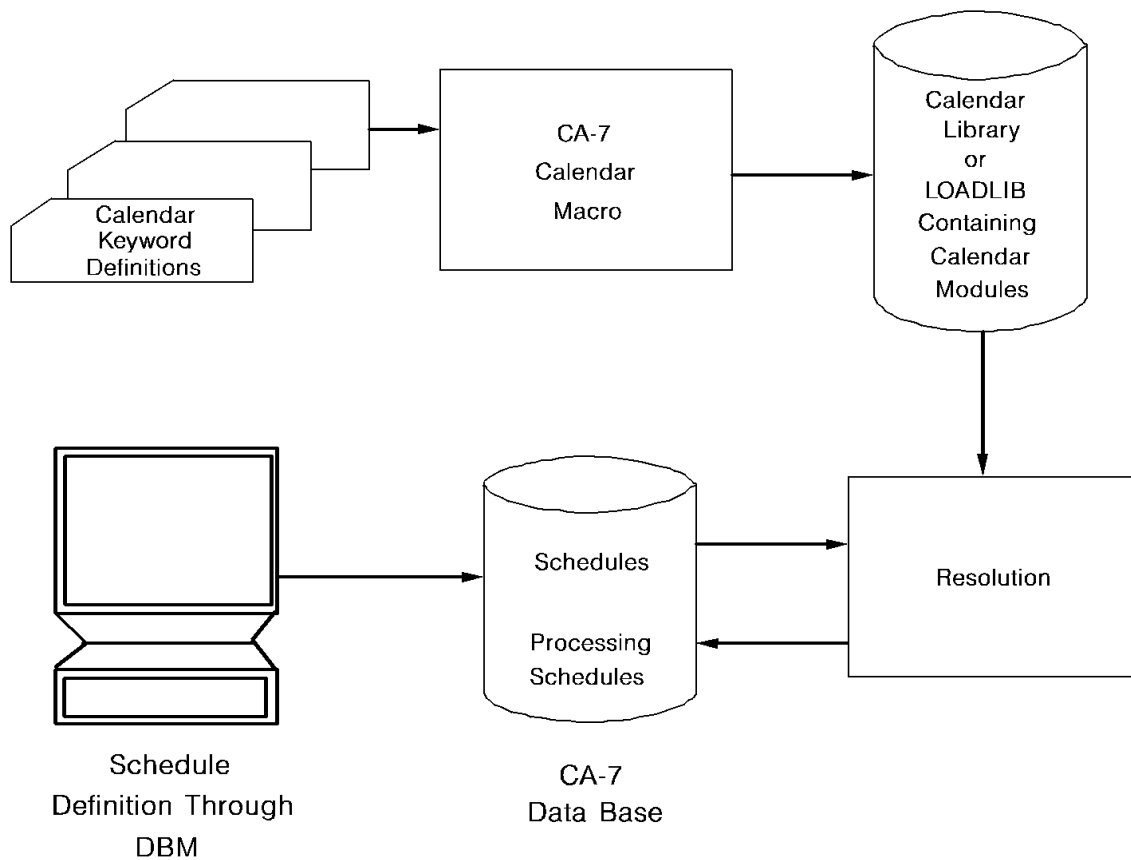


Figure 3-1. Calendar and Schedule Creation and Resolution

3.1.2 Schedule Modifications

Schedule Modifications apply to both jobs and input networks and can only be made to an already defined and resolved schedule. Such modifications are handy when a request is made to run a scheduled job or input network on a date and/or time that deviates from the existing schedule.

One way to make such changes to an existing schedule is to use the Schedule Modification DB.2.7 screen. This screen displays a SCHID in calendar format showing when the job or network does or does not run. The schedule may be temporarily modified until the next DB.2.7 screen or RESOLV command is used against it.

Another way to permanently alter a schedule is by using the DB.2 screen to change it and then issuing a RESOLV command.

3.1.2.1 Schedule Scan

Schedule scan causes scheduled work to be automatically brought into the system. Based on time intervals specified in the SCHEDULE statement in the initialization file, schedule scan is activated and loads work into the system which is to start within the specified interval. After a system down condition, schedule scan normally continues where it left off; however, under unusual conditions it may be set back to a specific time to allow for easier recovery.

3.1.3 Event Scheduling

3.1.3.1 Triggers

Schedule triggers can be used as a more efficient alternative to date/time schedules. Triggers cause jobs to be dynamically scheduled based solely on completion of an event. The triggering event may be date/time scheduled or it too can be automatically triggered by even another event. Schedule triggers have no association with a calendar. Therefore, no calendar related maintenance is ever required. Once defined, triggers are perpetual, remaining in effect until they are either deleted or modified by the user. In addition to relieving the user of annual calendar schedule maintenance, triggers are a far more efficient scheduling mechanism in terms of system overhead. There are three types of triggers available in CA-7:

- Job
- Network
- Data Set

With each trigger, multiple jobs can be triggered if so desired.

Job Triggers: When a predecessor job completes successfully, a trigger initiates scheduling of the successor or dependent job(s) that will be brought into the request queue. This newly triggered job may also have a trigger definition that, upon successful completion, causes another job to start. For ease of definition and understanding, the predecessor/successor relationship is commonly used to control sequencing of jobs in the production environment. Job triggers are defined with the DB.2.4 function.

Network Triggers: Input (or preprocessing) workstation networks can be used to trigger CPU jobs which are dependent on the completion of that network. These triggers take effect whenever the last workstation in the network is posted as being complete. The job(s) to be triggered is defined through the DB.2.5 function.

Multiple jobs can be triggered by the completion of a single network. Output (or postprocessing) networks cannot be used to trigger jobs.

Data Set Triggers: Data set output activity can also be used as a trigger. Whenever a sequential data set is created or updated, CPU jobs can be triggered by the completion of that activity. Scheduling of jobs in this manner is accomplished through the DB.2.6 and DB.6 screens.

A data set cannot be used as a trigger if it is defined as PERM=YES on the DB.6 screen or if it is output by a job which is defined as MAINT=YES.

Data set triggers are very useful if you have a job which needs to run only if a specific data set has been created.

If the data set used as a trigger specifies YES for the DB.6 screen parameter POST AT CLOSE TIME, the data set trigger takes effect when the data set is successfully closed after being opened for output or update. If POST AT CLOSE TIME is NO, then the trigger takes effect on successful completion of the job in which the data set was created or updated.

The advantage of posting at close time is that successful completion of an entire job is not necessary for data set triggering or requirement posting to take place.

Note: If you trigger a job at data set close, the trigger occurs then, even if the job that created the data set later abends. If the abended job is then restarted and the data set is re-created, the trigger occurs again.

3.1.4 On-Request Scheduling

Provisions are included in CA-7 to permit on-request scheduling of work. This is the method of scheduling jobs which do not have predefined processing cycles (or jobs that must be forced into early execution). However, on-request scheduling, using the RUN command, may optionally cause some of the normal scheduling activities to be bypassed. This includes verification of requirement availability, construction of postprocessing workstation networks, data set creation updates to the database and triggering of other data sets or jobs.

To accomplish on-request scheduling, use the DEMAND and RUN commands for jobs or the DMDNW and RUNNW commands for networks.

3.1.4.1 DEMAND Command

The DEMAND command can be used in either batch or online mode. Any job can be requested (brought into the CA-7 request queue) when the following is provided:

- A DEMAND (or DEMANDH) command
- Job name
- JCL library ID (only needed if the job is not already defined in the database)

The job is immediately brought into the CA-7 request queue. If the job has never been submitted by CA-7, a LOAD step is automatically inserted to build a database profile unless a SET=NDB parameter is used. The job then executes as it otherwise would without the LOAD step. This command is generally used to run a job on a onetime basis.

3.1.4.2 DMDNW Command

The DMDNW command can be used in either batch or online mode. Any workstation network in the database can be requested when the following is provided:

- A DMDNW command
- Network name

The network is immediately placed in a queue. If the network is defined for input, it is placed in the preprocessing queue. If the network is defined for output, it is placed in the postprocessing queue. When input networks are completed, job triggering and requirement posting occur.

3.1.4.3 RUN Command

The RUN command is used to request a job without verifying the availability of input requirements or performing the updates which normally follow successful job completion.

Use of the RUN or RUNH command causes a job to be brought into the request queue for scheduling. If workload balancing is being used, the job runs in CA-7 class 9.

The following normal scheduling activities of CA-7 are bypassed:

- Verification of input requirement availability
- Construction of postprocessing workstation networks
- Database index entry updates for newly created output data sets
- Prior-run queue updates
- Scheduling of job or data set triggered jobs
- Requirements are not satisfied for normally scheduled jobs
- Updating of database "last run" value

The following normal scheduling activities occur:

- The appropriate LTERM is prompted if the job becomes late.
- The appropriate LTERM is notified when the job completes.
- The RUNLOG is updated.

3.1.4.4 RUNNW Command

The RUNNW command is used to request a workstation network similar to the DMDNW command. However, input networks requested with the RUNNW command do not post job requirements or schedule job triggers.

Use of the RUNNW command causes a network to be placed in a queue immediately. If the network is defined for input, it is placed in the preprocessing queue. If the network is defined for output, it is placed in the postprocessing queue.

3.2 Defining Schedules

3.2.1 DB.2 - Scheduling Menu Screen

Use the DB.2 Menu screen to access to the scheduling definition formatted screens.

```

----- CA-7 SCHEDULING MENU -----
FUNCTION ==> xxxxxxxx

DATE/TIME SCHEDULING FOR:
  1 - CPU JOB
  2 - INPUT NETWORK
  3 - OUTPUT NETWORK

TRIGGER SCHEDULING FOR:
  4 - JOB TRIGGERING OTHER CPU JOB(S)
  5 - INPUT NETWORK TRIGGERING CPU JOB(S)
  6 - DATA SET TRIGGERING CPU JOB(S)

OTHER FUNCTIONS AVAILABLE:
  7 - MODIFICATION TO RESOLVED SCHEDULE DATES
  8 - BASE CALENDAR MAINTENANCE

PROGRAM: SM70 MSG-INDX: nn  -- DB.2    -- yy.ddd / hh:mm:ss
MESSAGE:

```

To display, enter:

- 2 as the FUNCTION on the DB Menu screen.
- DB.2 as the FUNCTION value on any other menu or formatted input screen.
- DB.2 or SCHD as a top line command.

To exit:

- Press the PF3 key to return to the DB Menu screen except with the TSO/ISPF interface when PF3 has been assigned to some other function.
- Move the cursor to the top line of the screen and enter the desired top line command if yet another function is desired.

3.2.1.1 Usage Notes

Select the desired function by entering the appropriate FUNCTION value and pressing the Enter key.

3.2.2 DB.2.1 - CPU Job Scheduling Screen

Use this screen to define or review options taken for CPU jobs with date/time schedules.

```

----- CA-7 CPU JOB SCHEDULING -----
FUNCTION: xxxxxxxx (CLEAR,DELETE,EDIT,FE,FETCH,REPL,RESOLV,SAVE)

JOB: xxxxxxxx

SCAL: xx (DEFAULT SCAL ID FOR ADDS)

SCHID-COUNT: nnn

PROGRAM: SM71 MSG-INDX: nn -- DB.2.1 -- yy.ddd / hh:mm:ss
MESSAGE:

```

To display, enter:

- 1 as the FUNCTION on the DB.2 Menu screen.
- 2.1 as the FUNCTION on the DB Menu screen.
- DB.2.1 as the FUNCTION value on any other menu or formatted input screen.
- DB.2.1 or SCHD,JOB as a top line command.

To exit:

- Press the PF3 key to return to the DB.2 Menu screen except with the TSO/ISPF interface when PF3 has been assigned to some other function.
- Move the cursor to the top line of the screen and enter the desired top line command if yet another function is desired.

3.2.2.1 Field Descriptions

FUNCTION:	The function to be performed. Value must be the name of some other screen or one of the following:
CLEAR	Clears the screen input data fields and resets the SCHID-COUNT to 0.
DELETE	Deletes a job schedule member from the database.
EDIT	Transfers the user to the EDIT facility as shown on the CPU Job Scheduling Parameter Edit screen, which allows the addition or modification of job schedules in a work area.
FE	A combination of the FETCH and EDIT commands. Retrieves all schedule data currently defined for the job and transfers you to the secondary scheduling screen. The scheduling information for the first schedule ID defined for the job is displayed.
FETCH	Retrieves job schedule data from the database and makes all schedule information for this job available to the user through the EDIT function of this screen.
REPL	Replaces existing schedule member in the database. Any SCHMODs that have been done to the schedule are cleared. This is not the same as overlaid, which can only occur when a RESOLV is done.
RESOLV	Generates a RESOLV command for the job indicated using the calendar specified. Any new or modified calendar based schedule must be RESOLVed before it takes effect.
SAVE	Adds a new job schedule member to the database for this job.
	Note: The FETCH function may reference one job name while the SAVE or REPL function specifies a different job name. This allows the user to make some changes to the schedule of an existing job and save those changes as the schedule for a new job.
	Required: Yes
	Batch keyword: Positional parameter
n/a	This field has a constant value of JOB for jobs.
	Required: Yes for batch only
	Batch keyword: Positional parameter

JOB:	The required job name for which schedule information is being defined.
Size/Type:	1 to 8 alphanumeric characters
Required:	Yes
Batch keyword:	JOB
SCAL:	This field indicates the last 2 characters of the default base calendar ID to be used for schedule resolution. The complete calendar ID used is in the format SCALnnxx where nn is the current year and xx is the value entered. This value is used for all SCHIDs that do not have a specific value coded on the DB.2.1-E screen.
Size/Type:	2 alphanumeric characters
Required:	Yes - for the SAVE, REPL, and RESOLV functions No - for CLEAR, DELETE, and EDIT functions Automatically supplied for FETCH or FE functions
Batch keyword:	SCAL
SCHID-COUNT:	A system-generated field which tells the user how many schedule IDs currently exist in the Edit Work File.

3.2.2.2 Usage Notes

You must resolve schedules defined on this screen against the specified base calendar with the RESOLV function or top line command before CA-7 automatic scheduling can commence.

3.2.3 CPU Job Scheduling Parameter Edit Screen

Use this screen to define or review schedules for jobs which are run on a date and time basis.

```

----- CA-7 CPU JOB SCHEDULING PARAMETER EDIT -----
FUNCTION: xxxxxxxx (ADD,DELETE,EXIT,FORMAT,LIST,REPL,SAVE,SR,SS)
JOB: xxxxxxxx SCHID: nnn SCAL: xx ROLL: x INDEX: nnnn
      DOTM LDTM SBTM
      hhmm hhmm hhmm

-- _ -- DAILY
-- _ -- WEEKLY  SUN: _ MON: _ TUE: _ WED: _ THU: _ FRI: _ SAT: _

-- _ -- MONTHLY JAN: _ FEB: _ MAR: _ APR: _ MAY: _ JUN: _
                JUL: _ AUG: _ SEP: _ OCT: _ NOV: _ DEC: _
      WEEK: _____ DAY-OF-WEEK: _____
      RDAY: _____

-- _ -- ANNUAL  DAY: _____
-- _ -- SYMETRIC START: ____ SPAN: ____
                                     DEFAULT SCAL: xx
                                     SCHID-COUNT: nnn

PROGRAM: SM72 MSG-INDX: nn -- DB.2.1-E -- yy.ddd / hh:mm:ss
MESSAGE:

```

To display, enter:

- EDIT as the function on the DB.2.1 screen.

To exit:

- Press the PF3 key to return to the DB.2 Menu screen except with the TSO/ISPF interface when PF3 has been assigned to some other function.
- Move the cursor to the top line of the screen and enter the desired top line command if yet another function is desired.

3.2.3.1 Field Descriptions

FUNCTION:	The function to be performed. Value must be the name of some other screen or one of the following:
ADD	Adds a new SCHID to the work area.
DELETE	Deletes a SCHID from the work area.
EXIT	Returns the user to the DB.2.1 screen restoring the SCHID-COUNT and schedule data to that which existed when EDIT was initially entered and clears the work area. Any changes made are ignored.
FORMAT	Clears the screen of user input data.
LIST	Lists all existing SCHIDs, one per screen.
REPL	Replaces an existing SCHID in work area. (Before replacing, you must use the LIST command to display the existing record.)
SAVE	Updates SCHID-COUNT and SCHID information, and returns the user to the DB.2.1 screen. Work area data (changes) are retained but the database is not updated.
SR	Combination function which results in a SAVE, a return to the DB.2.1 screen, and a subsequent REPL to replace an existing schedule in the database. Saves all schedule IDs into the Active Area and replaces them in the CA-7 database. If not specified, the scheduling changes are lost once leaving this screen. Entering the SR function automatically returns you to the prior scheduling screen.
SS	Combination function which results in a SAVE, a return to the DB.2.1 screen, and a subsequent SAVE to add a newly defined schedule in the database.
	Required: Yes
	Batch keyword: Positional parameter
JOB	This is informational only. It reflects the job name from the previous screen.
SCHID:	The numeric schedule ID on which the user wishes to perform the specified action.
	Size/Type: 1 to 3 numeric characters from 1 through 255
	Required: Yes
	Batch keyword: SCHID

SCAL	<p>The override of the default SCAL for this SCHID. If this SCHID is using the default, this field is blank.</p> <p>Size/Type: 2 alphanumeric characters</p> <p>Required: No</p> <p>Batch keyword: SCAL</p>
ROLL:	<p>Specifies the desired action to be taken when a schedule day falls on a base calendar nonavailable processing day. This is not used if the DAILY option is used. If used, the value must be one of the following:</p> <p>B Roll the schedule back to the previous available processing day in the Base Calendar.</p> <p>F Roll the schedule forward to the next available processing day.</p> <p>N Do not roll. Schedule day must stand.</p> <p>D Do not roll and do not schedule.</p> <p>Default: D</p> <p>Required: No</p> <p>Batch keyword: ROLL</p>
INDEX:	<p>Specifies an adjustment to schedule days. After exercising the ROLL option, the schedule is adjusted, forward for plus or backward for minus, by the number of working days entered.</p> <p>Size/Type: 4 numeric characters specified as Innn where I can be plus (unsigned) or minus (-) and nnn can be 0 to 365 days</p> <p>Required: No</p> <p>Batch keyword: INDEX</p> <p>You can use this field together with the RDAY field to schedule a job to run 3 work days before the 15th of the month: specify -3 in the INDEX field, X in the MONTHLY field, and 15 in the RDAY field.</p>
DOTM	<p>The due-out time of day for this schedule ID.</p> <p>Size/Type: 4 numeric characters specified as hhmm where hh can be 00 through 24 and mm can be 00 through 59</p> <p>Required: Yes - for ADD and REPL functions No - for DELETE, EXIT, FORMAT, LIST, SAVE, SR, and SS functions</p> <p>Batch keyword: TIME</p> <p>In batch format, this field and the next two fields must be provided as a list of hhmm values as follows:</p> <p>TIME=(dueout,lead,submit)</p>

LDTM	<p>The lead time (elapsed or clock time) for this schedule ID. Lead time specifies the amount of time necessary to ensure that the job completes before its due-out time plus any additional time required to perform setup functions. This value is used for deadline prompting and forecasting only if elapsed (clock) time on the DB.1 screen is set to 2359.</p> <p>Size/Type: 4 numeric characters specified as hhmm where hh can be 00 through 24 and mm can be 00 through 59</p> <p>Required: Yes - for ADD and REPL functions No - for DELETE, EXIT, FORMAT, LIST, SAVE, SR, and SS functions</p> <p>Batch keyword: TIME</p>
SBTM	<p>The submit time of day for this schedule ID. If specified, the job is not submitted before this time. If the submit time is before deadline start time, the submit time requirement will be automatically satisfied when the job enters the queue. A submit time of zeros is the same as not specifying it, and no submit time is set up.</p> <p>Size/Type: 4 numeric characters specified as hhmm where hh can be 00 through 24 and mm can be 00 through 59</p> <p>Required: No</p> <p>Batch keyword: TIME</p>
DAILY	<p>Indicates that the user wishes to define a daily schedule. Daily means every available processing day as defined by the Base Calendar. This is selected by entering an X or Y. If DAILY is used, the ROLL function has no effect.</p> <p>Required: No</p> <p>Batch keyword: DAILY</p>
WEEKLY	<p>Indicates that the user wishes to define a weekly schedule. This is selected by entering an X or Y. If WEEKLY is used, the run days of the week must be selected using the SUN through SAT fields.</p> <p>Required: No</p> <p>Batch keyword: WEEKLY</p>

SUN: thru SAT: These fields are used to define specific days of the week on which the job is to be run.

Online: Place an X in screen field names SUN through SAT as necessary to indicate the weekly run days.

Required: Yes (if WEEKLY is used)

Batch keyword: DAYS

Batch: Place an X in any of the 7 positions following the batch keyword DAYS to indicate weekly run days. A day is skipped by omitting an X for that day. For example, if a job is to run weekly on Sunday, Wednesday and Saturday, the user would specify:

DAYS=(X,,,X,,,X)

MONTHLY Indicates that the user wants to define a monthly schedule. If this field is used, the user may optionally specify on which particular months the job is to run. If specific months are not specified in the JAN through DEC fields, all months are assumed. Selection is indicated by entering an X or Y.

At least WEEK(S) and DAY-OF-WEEK (DOW) or RDAY(S) must also be specified if MONTHLY is used.

Required: No

Batch keyword: MONTHLY

JAN: thru DEC: Online: Place an X in screen field names JAN through DEC to indicate the run months during the year. Used with MONTHLY.

Default: Every month if MONTHLY is used and no months are chosen

Required: No

Batch keyword: MONTHS

Batch: Place an X in any of the 12 positions following the batch keyword MONTHS to indicate run months. A month is skipped by omitting an X for that month. For example, if a job is to run in January, June, July and November, the user would specify:

MONTHS=(X,,,,,X,X,,,X,)

WEEK:

Specifies which weeks of the month the job is to run. The values specified can be positive (unsigned), negative (-) or slash (/).

Positive values 1, 2, 3, 4, or 5 are used to indicate days of week relative to the beginning of the month. Negative values -0, -1, and -2 are used to indicate occurrence of day of week relative to the end of the month. Slashes indicate NOT the following value. For example, a job that is to run on the first occurrence of day of week of every month and is also to run on the last occurrence of day of week of the month would be entered as:

Online:

WEEK:1,-0 (separated by blanks or commas)

Size/Type: 1 to 14 numeric characters

Required: Yes (if MONTHLY and DAY-OF-WEEK are used)

Batch keyword: WEEKS

Batch:

WEEKS=1 -0 (separated by blanks)

DAY-OF-WEEK:

Indicates which day(s) of the week the job is to be run. If used, each day must be the first three letters of the desired run day(s) just as it appears on the screen following the weekly field (for example, SUN, MON, and so forth).

Online:

DAY-OF-WEEK:TUE,THU

(separated by blanks or commas)

Size/Type: 3 to 27 alphanumeric characters

Required: Only if WEEK is used with MONTHLY

Batch keyword: DOW

Batch:

DOW=TUE THU (separated by blanks)

RDAY:

Specifies relative days of the month on which the job is to run. Used with MONTHLY. A day relative to the beginning or end of the month is specified. If a positive (unsigned) value is used, the job runs that day relative to the first of the month. Negative (-) values are used to indicate run days relative to the end of the month.

Additionally, if there are days of the month when this job is not to run, a slash (/) may be used with the unsigned or negative values. Valid positive values range from 1 to 31. Valid negative values range from -0 to -30.

Size/Type: 1 to 60 numeric characters

Required: No

Batch keyword: RDAY

To understand the meaning and use of RDAY, assume a job is to run daily, but not on the first and last day of each month. The user could select DAILY, MONTHLY and specify:

Online:

RDAY:/1,-0 (separated by a blank or comma)

Batch:

RDAYS=/1 /-0 (separated by a blank)

Note: RDAY(S) represents calendar days, unless the base calendar was generated with OPTIONS=SCHDYONLY, in which case RDAY(S) represents processing days.

ANNUAL

Defines an annual schedule. The user selects this type of schedule by placing an X or Y in this optional field. If ANNUAL is used, DAY is required.

Required: No

Batch keyword: ANNUAL

DAY:

Indicates on which days of the annual schedule the user wishes to run the job. Days are entered as 1 through 366 and are prefixed by positive (unsigned) or slash (/) values. Slash indicates NOT this day.

Size/Type: 1 to 55 numeric characters

Required: Yes (if ANNUAL is used)

Batch keyword: ANNDAYS

For example, if a job is to run on days 121, 132, 240, and 241, but is not to run on days 122, 242, and 1, the user would specify:

Online:

DAY:/1,121,/122,132,240,241,/242
(separated by blanks or commas)

Batch:

ANNDAYS=/1 121 /122 132 240 241 /242
(separated by blanks)

By default, days not specifically defined as processing days (with a number) are considered as unscheduled days unless another field, such as WEEKLY or MONTHLY schedules those days. The use of the slash (/) examples given above is only provided here to illustrate the format of the date specification.

The following illustrates a more practical example of using "not days" (/). Assume a job's schedule has been defined as every Friday by selecting WEEKLY and FRI. During the week containing Julian day 177 (a Friday), the job is to run on Thursday (176) instead of Friday. The user would select WEEKLY, FRI, ANNUAL, and specify:

Online:

DAY:/177,176

Batch:

ANNDAYS=/177 176

When specifying annual days in a schedule definition, keep in mind whether this schedule is to be resolved for a January through December schedule or a July through June schedule. For January through December resolutions, the Julian dates specified are simply for the year resolved.

For July through June resolutions, July through December dates are for the current year. January through June dates are for the next year.

DEFAULT SCAL:

This reflects the default SCAL value from the previous screen. This value applies to all SCHIDs unless the SCAL field is supplied on the current screen.

SYMETRIC

This field is used in combination with the START and SPAN fields to define a symmetric schedule. With this option, schedules are defined as beginning on the date specified with the START field and recurring every nnn days as specified with the SPAN field. (This field is misspelled intentionally to provide compatibility with the batch keyword which cannot exceed 8 characters.) The selection of this type of schedule is made by entering an X or Y.

Required: No

Batch keyword: SYMETRIC

START:

This field is used in combination with SPAN and SYMETRIC to define a symmetric schedule. This field is required when the SYMETRIC option is taken. Value must be specified as the Julian day of the year on which the symmetric schedule is to begin. This value should be evaluated yearly before schedule resolution and may need to be changed each year.

If the schedule will be resolved on a July-June basis, the start date must be within the first of the two years in which the schedule overlaps. For January-December schedules, it is simply relative to January 1 of the year specified on a RESOLV command.

If the calendar against which this SCHID will be resolved does not have the OPTIONS=SCHDYONLY, the START day will be adjusted each year to maintain the job's symmetric schedule from the job's original specified START day.

If the calendar against which the SCHID will be resolved was generated with overall available processing days only (OPTIONS=SCHDYONLY), the start date must specify the first day on which the job would actually run. This requires manually setting (verifying) the START value each year before the RESOLV.

If a nonprocessing day is specified, the start day is the next processing day found in the calendar. For example, if January 1 is a nonprocessing day and the calendar was generated with OPTIONS=SCHDYONLY, and 1 is specified as the START day, January 2 is the actual START day (provided it is an available processing day). SPAN is based from January 2 in this example rather than January 1.

For other types of calendars, the start date is determined at RESOLV time based on the ROLL and INDEX options taken.

Size/Type: 3 numeric characters specified as Julian day of the year from 1 to 365

Required: No

Batch keyword: START

SPAN: This field is used in combination with SYMETRIC and START to define symmetric schedules. This field is required if the SYMETRIC option is taken. When used, specifies the number of days between scheduled processing cycles.

If the calendar against which the SCHID will be resolved was generated with processing days only (OPTIONS=SCHDYONLY), the SPAN value is specified as the number of available on processing days off between and including the next processing date as opposed to the actual number of days. With this type of calendar, the ROLL and INDEX options serve no practical purpose. For other calendar types, the SPAN value is specified as the number of calendar days between processing cycles and the ROLL and INDEX options can be used as necessary.

Size/Type: 1 to 3 numeric characters from 1 to 255

Required: No

Batch keyword: SPAN

SCHID-COUNT: A system-generated field which tells the user the current schedule ID count for this job.

3.2.3.2 Batch Input Example

This batch input example adds schedule ID 1 for job CA07XX01 for a daily schedule, using calendar 03, with a due-out time of 0800, and a lead time of 1 hour.

```
SCHD
JOB
EDIT
ADD,SCHID=1,DAILY=Y,TIME=(0800,0100)
SAVE
SAVE,JOB=CA07XX01,SCAL=03
DBM
```

3.2.4 DB.2.2 - Input Network Scheduling Screen

Use this screen to define or review scheduling options for input networks.

```

----- CA-7 INPUT NETWORK SCHEDULING -----
FUNCTION: xxxxxxxx (CLEAR,DELETE,EDIT,FE,FETCH,REPL,RESOLV,SAVE)

NETWORK: xxxxxxxx

SCAL: xx (DEFAULT SCAL ID FOR ADDS)

SCHID-COUNT: nnn

PROGRAM: SM71 MSG-INDX: nn -- DB.2.2 -- yy.ddd / hh:mm:ss
MESSAGE:

```

To display, enter:

- 2 as the FUNCTION on the DB.2 Menu screen.
- 2.2 as the FUNCTION on the DB Menu screen.
- DB.2.2 as the FUNCTION on any other menu or formatted input screen.
- DB.2.2 or SCHD,INWK as a top line command.

To exit:

- Press the PF3 key to return to the DB.2 Menu screen except with the TSO/ISPF interface when PF3 has been assigned to some other function.
- Move the cursor to the top line of the screen and enter the desired top line command if yet another function is desired.

3.2.4.1 Field Descriptions

FUNCTION:	The function to be performed. Value must be the name of some other screen or one of the following:
CLEAR	Clears the screen input data fields and resets the SCHID-COUNT to 0.
DELETE	Deletes an input network schedule member from the database.
EDIT	Transfers the user to the EDIT facility as shown on the Input Network Scheduling Parameter Edit screen. This allows the addition or modification of input network schedules in a work area.
FE	Combination of FETCH and EDIT.
FETCH	Retrieves input network schedule data from the database and makes all of the schedule information for this input network available to the user through the EDIT function of this screen.
REPL	Replaces an existing schedule in the database.
RESOLV	Generates a RESOLV command for the network indicated using the calendar specified. Any new or modified calendar based schedule must be RESOLVED before it takes effect.
SAVE	Adds a new schedule to the database for this input network. The user may SAVE to any input network name but the number of stations specified on the EDIT screen for the network must equal the number defined for the network.
	Required: Yes
	Batch keyword: Positional parameter
n/a	This field has a constant value of INWK for input networks.
	Required: Yes for batch only
	Batch keyword: Positional parameter
NETWORK:	The input network name for which schedule information is being defined.
	Size/Type: 1 to 8 alphanumeric characters
	Required: Yes
	Batch keyword: NETWORK

SCAL: This field indicates the last 2 characters of the default base calendar ID to be used for Schedule Resolution. The complete calendar ID used is in the format SCALnnxx where nn is the current year and xx is the value entered. This value is used for all SCHIDs that do not have a value coded on the DB.2.2-E screen.

Size/Type: 2 alphanumeric characters

Required: Yes - for SAVE, REPL, and RESOLV functions
No - for CLEAR, DELETE, EDIT, FE, and FETCH functions

Batch keyword: SCAL

SCHID-COUNT:

A system-generated field which shows how many schedule IDs currently exist for this network.

3.2.5 Input Network Scheduling Parameter Edit Screen

Use this screen to define or review scheduling parameters for input networks which are to be scheduled on a date and time basis.

```

----- CA-7 INPUT NETWORK SCHEDULING PARAMETER EDIT -----
FUNCTION: xxxxxxxx (ADD,DELETE,EXIT,FORMAT,LIST,REPL,SAVE,SR,SS)
NWK: xxxxxxxx SCHID: nnn SCAL: xx ROLL: x INDEX: nnnn
STATION DOTM LDTM DODY STATION DOTM LDTM DODY STATION DOTM LDTM DODY
1 xxxxxxxx hhmm hhmm nnnn 2 xxxxxxxx hhmm hhmm nnnn 3 xxxxxxxx hhmm hhmm nnnn
4 xxxxxxxx hhmm hhmm nnnn 5 xxxxxxxx hhmm hhmm nnnn 6 xxxxxxxx hhmm hhmm nnnn
7 xxxxxxxx hhmm hhmm nnnn 8 xxxxxxxx hhmm hhmm nnnn 9 xxxxxxxx hhmm hhmm nnnn

-- _ -- DAILY
-- _ -- WEEKLY SUN: _ MON: _ TUE: _ WED: _ THU: _ FRI: _ SAT: _

-- _ -- MONTHLY JAN: _ FEB: _ MAR: _ APR: _ MAY: _ JUN: _
JUL: _ AUG: _ SEP: _ OCT: _ NOV: _ DEC: _
WEEK: _____ DAY-OF-WEEK: _____
RDAY: _____

-- _ -- ANNUAL DAY: _____
-- _ -- SYMETRIC START: ____ SPAN: ____
                                     DEFAULT SCAL: xx
                                     SCHID-COUNT: nnn

PROGRAM: SM72 MSG-INDX: nn -- DB.2.2-E -- yy.ddd / hh:mm:ss
MESSAGE:

```

To display, enter:

- EDIT as the function on the DB.2.2 screen.

To exit:

- Press the PF3 key to return to the DB.2 Menu screen except with the TSO/ISPF interface when PF3 has been assigned to some other function.
- Move the cursor to the top line of the screen and enter the desired top line command if yet another function is desired.

3.2.5.1 Field Descriptions

FUNCTION:	The function to be performed. Value must be the name of some other screen or one of following:
ADD	Adds a new SCHID to the work area.
DELETE	Deletes a SCHID from the work area.
EXIT	Returns the user to the DB.2.2 screen and restores the SCHID-COUNT and schedule data to that which existed when EDIT was initially entered and clears the work area.
FORMAT	Clears the screen of user input data.
LIST	Lists all SCHIDs, one per screen.
REPL	Replaces an existing SCHID in the work area.
SAVE	Updates SCHID-COUNT and SCHID information and returns the user to the DB.2.2 screen. Work area data (changes) are retained but the database is not updated.
SR	Combination function which results in a SAVE, a return to the DB.2.2 screen, and a subsequent REPL to replace an existing schedule in the database. Saves all schedule IDs into the Active Area and replaces them in the CA-7 database. If not specified, the scheduling changes are lost once leaving this screen. Entering the SR function automatically returns you to the prior scheduling screen.
SS	Combination function which results in a SAVE, a return to the DB.2.2 screen, and a subsequent SAVE to add a newly defined schedule in the database.
	Required: Yes
	Batch keyword: Positional parameter
NWK	This is informational only. It reflects the network name from the previous screen.
SCHID:	The numeric schedule ID on which the user wishes to perform the specified action.
	Size/Type: 1 to 3 numeric characters from 1 to 255
	Required: No
	Batch keyword: SCHID
SCAL:	This is the override of the default SCAL for this SCHID. If this SCHID is using the default, this field is blank.
	Size/Type: 2 alphanumeric characters
	Required: No
	Batch keyword: SCAL

ROLL:	<p>Specifies the desired action to be taken when a schedule day falls on a base calendar nonavailable processing day. If used, the value must be one of the following:</p> <p>B Roll the schedule back to the previous available processing day in the Base Calendar.</p> <p>F Roll the schedule forward to the next available processing day.</p> <p>N Do not roll. Schedule day must stand.</p> <p>D Do not roll and do not schedule.</p> <p>Default: D</p> <p>Required: No</p> <p>Batch keyword: ROLL</p>
INDEX:	<p>A field used to specify an adjustment to schedule days. After exercising the ROLL option, the schedule is adjusted, forward or backward, by the number of working days entered.</p> <p>Size/Type: 1 to 4 numeric characters specified as Innn where I can be plus (unsigned) or minus (-) and nnn can be 0 to 365 days</p> <p>Required: No</p> <p>Batch keyword: INDEX</p>
STATION	<p>Network workstation names. Up to nine workstations are listed. Station names are listed in the sequence in which their tasks are to be performed (that is, as defined for the network). The same station name may appear multiple times in the list. This field is for station identification purposes only. For batch, see the note under the TIME batch keyword.</p> <p>Size/Type: 1 to 8 alphanumeric characters</p> <p>Required: No</p> <p>Batch keyword: Not applicable (see TIME)</p>
DOTM	<p>This is the due-out time of day for the network workstation using this schedule ID.</p> <p>Size/Type: 1 to 4 numeric characters specified as hhmm, where hh can be 00 through 24 and mm can be 00 through 59</p> <p>Required: Yes - for ADD or REPL functions No - for DELETE, EXIT, FORMAT, LIST, SAVE, SR, or SS functions</p> <p>Batch keyword: TIME</p> <p>Note for batch input:</p> <p>TIME=(dotm1,ldtm1,dody1,dotm2,ldtm2,dody2,...,dotmn,ldtmn,dodyn)</p> <p>must be used, where the numeric value (or n) corresponds to the relative number of the station.</p>

LDTM	<p>The lead time for the workstation for this schedule ID. This is the amount of elapsed time necessary to ensure that the station completes its task by the scheduled due-out time.</p> <p>Size/Type: 1 to 4 numeric characters specified as hhmm, where hh can be 00 through 24 and mm can be 00 through 59</p> <p>Required: Yes - for ADD or REPL functions No - for DELETE, EXIT, FORMAT, LIST, SAVE, SR, or SS functions</p> <p>Batch keyword: TIME</p>
DODY	<p>The due-out day for the workstation in the network for this schedule ID. It is always relative to the first workstation in the network. It indicates the number of days elapsed from the first workstation.</p> <p>Size/Type: 1 to 4 numeric characters from 0 through 255</p> <p>Default: 0</p> <p>Required: Yes (if the lead time (elapsed time) between workstations in a network causes the schedule to cross any midnight boundary)</p> <p>Batch keyword: TIME</p>
DAILY	<p>Indicates that the user wishes to define a daily schedule. Daily means every available processing day as defined by the Base Calendar. This is indicated by entering an X or Y.</p> <p>Required: No</p> <p>Batch keyword: DAILY</p>
WEEKLY	<p>Indicates that the user wishes to define a weekly schedule. This is selected by entering an X or Y. If WEEKLY is used, the run days of the week must be selected using the SUN thru SAT fields.</p> <p>Required: No</p> <p>Batch keyword: WEEKLY</p>
SUN: thru SAT:	<p>These fields are used to define specific days of the week on which the network is to be run.</p> <p>Online: Place an X in screen field names SUN through SAT to indicate the weekly run days.</p> <p>Required: Yes (if WEEKLY is used)</p> <p>Batch keyword: DAYS</p>

Batch: Place an X in any of the 7 positions following the batch keyword DAYS to indicate weekly run days. A day is skipped by omitting an X for that day. For example, if a network is to run weekly on Sunday, Wednesday and Saturday, the user would specify:

DAYS=(X,,,X,,,X)

MONTHLY

Indicates that the user wants to define a monthly schedule. If this field is used, the user may specify on which particular months the network is to run. If specific months are not specified in the JAN thru DEC fields, all months are assumed. Selection is indicated by entering an X or Y.

Required: No

Batch keyword: MONTHLY

At least WEEK(S) and DAY-OF-WEEK (DOW) or RDAY(S) must be specified if MONTHLY is used.

JAN: thru DEC:

Online: Place an X in screen field names JAN through DEC to indicate the run months during the year. Used with MONTHLY.

Default Every month if MONTHLY is used and no months are chosen

Required: No

Batch keyword: MONTHS

Batch: Place an X in any of the 12 positions following the batch keyword MONTHS to indicate run months. A month is skipped by omitting an X for that month. For example, if a network is to run in January, June, July and November, the user would specify:

MONTHS=(X,,,,,X,X,,,X,)

WEEK:

Specifies which weeks of the month the network is to run. Used with MONTHLY and DAY-OF-WEEK. The values specified can be positive (unsigned), negative (-) or slash (/).

Positive values 1, 2, 3, 4, or 5 are used to indicate which occurrence of day of week relative to the beginning of the month. Negative values -0, -1, -2, -3, or -4 are used to indicate which occurrence of day of week relative to the end of the month. Slashes indicate it is not the following value.

Size/Type: 1 to 14 numeric characters

Required: No

Batch keyword: WEEKS

For example, a network that is to run on the first week of every month and is also to run on the last week of the month, would be entered as:

Online:

WEEK:1,-0 (separated by blanks or commas)

Batch:

WEEKS=1 -0 (separated by blanks)

DAY-OF-WEEK:

Indicates which day of the week the network is to be run. Used with WEEK and MONTHLY. If used, must be the first three letters of the desired run day just as it appears on the screen following the weekly field (for example, SUN, MON, and so forth).

Online:

DAY-OF-WEEK:TUE,THU

(separated by blanks or commas)

Size/Type: 3 to 27 alphanumeric characters

Required: Yes (if WEEK is used)

Batch keyword: DOW

Batch:

DOW=TUE THU (separated by blanks)

RDAY:

This field is used to specify relative days of the month on which the network is to run. Used with MONTHLY. A day relative to the beginning or end of the month is specified. If a positive (unsigned) value is used, the network runs that day relative to the first of the month. Negative (-) values are used to indicate run days relative to the end of the month.

Additionally, if there are days of the month when this network is not to run, a slash (/) may be used with the unsigned or negative values. Valid positive values range from 1 to 31. Valid negative values range from -0 to -30.

Size/Type: 1 to 60 numeric characters

Required: No

Batch keyword: RDAY

To understand the meaning and use of RDAY, assume a network is to run daily, but not on the first or last day of each month. The user could select DAILY, MONTHLY and specify:

Online:

RDAY:/1,-0 (separated by a blank or comma)

Batch:

RDAYS=/1 -0 (separated by a blank)

Note: RDAY(S) represents calendar days, unless the base calendar was generated with OPTIONS=SCHDYONLY, in which case RDAY(S) represents processing days.

ANNUAL

Defines an annual schedule. The user selects this type of schedule by entering an X or Y in this field. If ANNUAL is used, DAY must also be specified.

Required: No

Batch keyword: ANNUAL

DAY:

Indicates on which days of the annual schedule the user wishes to run the network. Days are entered as 1 through 366 and are prefixed by positive (unsigned) or slash (/) values. Slash indicates NOT this day.

Size/Type: 1 to 55 numeric characters

Required: Yes (if the ANNUAL option above is chosen)

Batch keyword: ANNDAYS

For example, if a network is to run on days 121, 132, 240, and 241, but is not on days 122, 242, and 1, you would specify:

Online:

DAY:/1,121,/122,132,240,241,/242

(separated by blanks or commas)

Batch:

ANNDAYS=/1 121 /122 132 240 241 /242

(separated by blanks)

By default, days not specifically defined as processing days (with a number) are considered as unscheduled days unless another field such as WEEKLY or MONTHLY schedules those days. The use of the slash (/) examples given previously is only provided here to illustrate the format of the date specification.

The following illustrates a more practical example of using "not days" (/). Assume a network schedule has been defined as every Friday by selecting WEEKLY and FRI. During the week containing Julian day 177 (Friday), the network is to run on Thursday (day 176) instead of Friday. The user would select WEEKLY, FRI, ANNUAL and:

Online:

DAY:/177,176

Batch:

ANNDAYS=/177 176

When specifying annual days in a schedule definition, keep in mind whether this schedule will be resolved for a January through December schedule or a July through June schedule. For January through December resolutions, the Julian dates specified are simply for the year resolved.

For July through June resolutions, July through December dates are for the current year. January through June dates are for the next year.

DEFAULT SCAL:

This reflects the default SCAL value from the previous screen. This value applies to all SCHIDs unless the SCAL field is supplied on the current screen.

SYMETRIC

This field is used with the START and SPAN fields to define a symmetric schedule. (This field is misspelled intentionally on the screen to provide compatibility with the batch keyword which cannot exceed 8 characters in length.) The selection is made by entering an X or Y. With this option, schedules are defined as beginning on the date specified with the START field and recurring every nnn days as specified in the SPAN field.

Required: No
Batch keyword: SYMETRIC

START:

This field is used with the SYMETRIC field to set a starting day.

Size/Type: 1 to 3 numeric characters specified as Julian day from 1 to 365
Required: Yes (if SYMETRIC option is chosen)
Batch keyword: START

If the schedule will be resolved on a July-June basis, the start date must be within the first of the two years in which the schedule overlaps. For January-December schedules, it is simply relative to January 1 of the year specified on a RESOLV command.

If the calendar against which this SCHID will be resolved does not have the OPTIONS=SCHDYONLY, the START day will be adjusted each year to maintain the job's symmetric schedule from the job's original specified START day.

If the calendar against which the SCHID will be resolved was generated with overall available processing days only (OPTIONS=SCHDYONLY), the start date must specify the first day on which the network would actually run. This requires manually setting (verifying) the START value each year before the RESOLV.

If a nonprocessing day is specified, the start day is the next processing day found in the calendar. For example, if January 1 is a nonprocessing day and the calendar was generated with OPTIONS=SCHDYONLY, and 1 is specified as the START day, January 2 is the actual START day (provided it is an available processing day). SPAN is based from January 2 in this example rather than January 1.

For other types of calendars, the start date is determined at RESOLV time based on the ROLL and INDEX options taken.

SPAN:

This field is used in combination with SYMETRIC and START to define symmetric schedules. When used, specifies the number of days between scheduled processing cycles.

If the calendar against which the SCHID will be resolved was generated with processing days only (OPTIONS=SCHDYONLY), the SPAN value is specified as the number of available processing days between and including the next processing date as opposed to the actual number of days. With this type of calendar, the ROLL and INDEX options serve no practical purpose. For other calendar types, the SPAN value is specified as the number of calendar days between processing cycles and the ROLL and INDEX options can be used as necessary.

Size/Type: 1 to 3 numeric characters no greater than 255

Required: Yes (if the SYMETRIC option is chosen)

Batch keyword: SPAN

SCHID-COUNT:

A system-generated field which tells the user the current schedule ID count for this network.

3.2.5.2 Batch Input Example

This batch input example adds schedule ID 1 for input network TESTINNW as a DAILY schedule using calendar PE. There are two stations in this network. The first station has a due-out time of 0800 with a lead time of 1 hour. The second station has a due-out time of 0900 with a lead time of 30 minutes.

```
SCHD
INWK
EDIT
ADD,SCHID=1,DAILY=Y,TIME=(0800,0100,,0900,0030)
SAVE
SAVE,NETWORK=TESTINNW,SCAL=PE
DBM
```

3.2.6 DB.2.3 - Output Network Scheduling Screen

Use this screen to define or review options taken for output network schedules.

```

----- CA-7 OUTPUT NETWORK SCHEDULING -----
FUNCTION: xxxxxxxx (CLEAR,DELETE,EDIT,FE,FETCH,REPL,RESOLV,SAVE)

NETWORK: xxxxxxxx

SCHID-COUNT: nnn

PROGRAM: SM71 MSG-INDX: nn  -- DB.2.3  -- yy.ddd / hh:mm:ss
MESSAGE:

```

To display, enter:

- 3 as the FUNCTION on the DB.2 Menu screen.
- 2.3 as the FUNCTION on the DB Menu screen.
- DB.2.3 as the FUNCTION on any other menu or formatted input screen.
- DB.2.3 or SCHD,ONWK as a top line command.

To exit:

- Press the PF3 key to return to the DB.2 Menu screen except with the TSO/ISPF interface when PF3 has been assigned to some other function.
- Move the cursor to the top line of the screen and enter the desired top line command if yet another function is desired.

3.2.6.1 Field Descriptions

FUNCTION:	The function to be performed. Value must be the name of some other screen or one of following:
CLEAR	Clears the screen input data fields and resets the SCHID-COUNT to 0.
DELETE	Deletes a network schedule member from the database.
EDIT	Transfers the user to the EDIT facility, as shown on the Output Network Scheduling Parameter Edit screen. This allows the addition or modification of network schedules in a work area.
FE	Combination of FETCH and EDIT.
FETCH	Retrieves output network data from the database and makes all of the schedule ID information for this output network available to the user through the EDIT function of this screen.
REPL	Replaces existing schedule member in the database.
RESOLV	Not applicable.
SAVE	Adds a new schedule to the database for this output network. The user may SAVE to any output network name but the number of stations in the EDIT screen for the network must equal the number defined for the network.
	Required: Yes
	Batch keyword: Positional parameter
n/a	This field has a constant value of ONWK for output networks.
	Required: Yes for batch only
	Batch keyword: Positional parameter
NETWORK:	The output network name for which schedule information is being defined.
	Size/Type: 1 to 8 alphanumeric characters
	Required: Yes
	Batch keyword: NETWORK
SCHID-COUNT:	This is a system-generated field which shows the user the current schedule ID count for this network.
	Note: There is no SCAL field for output networks since their schedule is actually based on the schedule of a connected CPU job. Also, the RESOLV function has no effect on output network schedules.

3.2.6.2 Usage Notes

If the network is connected to the CPU job with a DB.3.5 function, the network is scheduled automatically whenever the connected CPU job is scheduled. However, the network must be scheduled with DB.2.3 so that time values may be given.

Optionally, for networks which direct the manual handling of SYSOUT type data, users of CA-Dispatch and CA-7 RPT can have those products automatically issue a DMDNW command for the network. In those cases, the network must not be connected to a CPU job with the DB.3.5 screen.

3.2.7 Output Network Scheduling Parameter Edit Screen

Use this screen to define or review individual workstation schedules in an output network. The output network's schedule cycles are based on the schedules of the jobs to which it is connected.

----- CA-7 OUTPUT NETWORK SCHEDULING PARAMETER EDIT -----

FUNCTION: xxxxxxxx (ADD,DELETE,EDIT,FORMAT,LIST,REPL,SAVE,SR,SS)

NWK: xxxxxxxx SCHID: nnn

STATION	DOTM	LDTM	DODY	STATION	DOTM	LDTM	DODY	STATION	DOTM	LDTM	DODY
1 xxxxxxxx	hhmm	hhmm	nnnn	2 xxxxxxxx	hhmm	hhmm	nnnn	3 xxxxxxxx	hhmm	hhmm	nnnn
4 xxxxxxxx	hhmm	hhmm	nnnn	5 xxxxxxxx	hhmm	hhmm	nnnn	6 xxxxxxxx	hhmm	hhmm	nnnn
7 xxxxxxxx	hhmm	hhmm	nnnn	8 xxxxxxxx	hhmm	hhmm	nnnn	9 xxxxxxxx	hhmm	hhmm	nnnn

SCHID-COUNT: nnn

PROGRAM: SM72 MSG-INDX: nn -- DB.2.3-E -- yy.ddd / hh:mm:ss

MESSAGE:

To display, enter:

- EDIT as the function on the DB.2.3 screen.

To exit:

- Press the PF3 key to return to the DB.2 Menu screen except with the TSO/ISPF interface when PF3 has been assigned to some other function.
- Move the cursor to the top line of the screen and enter the desired top line command if yet another function is desired.

3.2.7.1 Field Descriptions

FUNCTION:	The function to be performed. Value must be the name of some other screen or one of the following:
ADD	Adds a new SCHID to the work area.
DELETE	Deletes a SCHID from the work area.
EXIT	Returns the user to the DB.2.3 screen, restores the SCHID-COUNT and schedule data to that which existed when EDIT was initially entered and clears the work area.
FORMAT	Clears the screen of user input data.
LIST	Lists all existing SCHIDs, one per screen.
REPL	Replaces an existing SCHID in the work area.
SAVE	Updates SCHID-COUNT and SCHID information and returns the user to the DB.2.3 screen. Work area data (changes) are retained but the database is not updated.
SR	Combination function which results in a SAVE, a return to the DB.2.3 screen, and a subsequent REPL to replace an existing schedule in the database. Saves all schedule IDs into the Active Area and replaces them in the CA-7 database. If not specified, the scheduling changes are lost once leaving this screen. Entering the SR function automatically returns you to the prior scheduling screen.
SS	Combination function which results in a SAVE, a return to the DB.2.3 screen, and a subsequent SAVE to add a newly defined schedule in the database.
	Required: Yes
	Batch keyword: Positional parameter
NWK	This is informational only. It reflects the network name from the previous screen.
SCHID:	The numeric schedule ID on which the user wishes to perform the specified action.
	Size/Type: 1 to 3 numeric characters from 1 through 255
	Required: Yes
	Batch keyword: SCHID

STATION

1 thru 9 Network workstation names. Up to nine workstations are listed. Station names are listed in the sequence in which their tasks are to be performed. The same station name may appear multiple times in the list. This field is for station identification purposes only.

Size/Type: 1 to 8 alphanumeric characters
Required: No
Batch keyword: Not applicable (see TIME)

DOTM

The due-out time of day for the network workstations on this schedule ID.

Size/Type: 4 numeric characters specified as hhmm, where hh can be 00 through 24 and mm can be 00 through 59
Required: Yes - for ADD or REPL functions
No - for DELETE, EXIT, FORMAT, LIST, SAVE, SR, and SS functions
Batch keyword: TIME

Note for batch input:
TIME=(dotm1,ldtm1,dody1,dotm2,ldtm2,dody2,...,dotmn,ldtmn,dodyn)
must be used where the numeric value (or n) corresponds to the number of the station.

DOTM is used to dynamically calculate the actual due-out time for the network when it is brought into the queue. If the JOB field of the DB.5 screen definition matches the job name to which the network is connected, then these values are used as the actual due-out times and no adjustments are made. If the job's DOTM is greater than the network's DOTM, the network DOTM is incremented by 24 hours (that is, the next day).

If the job name field of the network definition does not match the job name to which the network is connected, the actual due-out time for the station(s) in the network is calculated as follows:

Station # DOTM Calculation

- 1 x+y(1)
- 2 x+y(2)+DOTM(2)-DOTM(1)
- 3 x+y(3)+DOTM(3)-DOTM(1)
-

Where:
x=Job's DOTM+network connection LDTM+LDTM(1)
y=DODY*2400 to calculate the day displacement.

LDTM

The lead time for this schedule ID. This is the amount of elapsed time necessary to ensure that the network completes its task before its scheduled due-out time.

Size/Type: 4 numeric characters specified as hhmm, where hh can be 00 through 24 and mm can be 00 through 59

Required: Yes - for ADD or REPL functions
No - for DELETE, EXIT, FORMAT, LIST, SAVE, SR, and SS functions

Batch keyword: TIME

DODY

The due-out day for this workstation in the network, relative to the due-out time for the job to which the network is connected.

Size/Type: 1 to 4 numeric characters from 0 through 255

Default: 0

Required: Yes (if the elapsed time between workstations in a network causes the schedule to roll over any midnight boundary)

Batch keyword: TIME

SCHID-COUNT:

A system-generated field which tells the user the current schedule ID count for this network.

3.2.7.2 Batch Input Example

This batch input example adds schedule ID 1 for output network TESTOTNW. Since this is an output network, no calendar or frequency data are allowed. This network has two stations where the first station has a due-out time of 0800 with a lead time of 1 hour. The second station has a due-out time of 0900 with a lead time of 30 minutes.

```
SCHD
ONWK
EDIT
ADD,SCHID=1,TIME=(0800,0100,,0900,0030)
SAVE
SAVE,NETWORK=TESTOTNW
DBM
```

3.2.8 Trigger Scheduling Screens

The database elements which can trigger the scheduling of a job are jobs, data sets, and input networks. The topics which follow identify the different types corresponding to these elements, their keywords, and how they cause triggering.

All trigger commands share common field requirements. Required fields for all trigger commands (batch or online) are identified in 3.2.11.2, “Field Descriptions” on page 3-47.

Uniqueness of the trigger relationship is determined by the triggering element, triggered job name, and schedule ID. When a triggering element is successfully completed/created, the triggered job (TRGD-JOB) enters the request queue. The TRGD-JOB assumes the schedule ID of the triggering element unless TRGID is specified, in which case the schedule ID is replaced by TRGID.

Up to 255 job name/schedule ID combinations may be triggered per triggering element, or one job can be triggered by up to 255 elements.

3.2.9 DB.2.4 - Job Triggering Screen

Use this screen to review or define jobs which trigger scheduling of other jobs when they successfully complete.

CA-7 JOB TRIGGERING										
FUNCTION: xxxxxxxx (FORMAT,LIST,UPD)								PAGE nnnn		
JOB: xxxxxxxx										
OPT	SCHID	TRGD-JOB	TRGID	DOTM	QTM	LDTM	SBTM	*----	EXCEPTIONS	----
x	nnn	xxxxxxx	nnn	hhmm	mmmm	hhmm	hhmm	xxxxxxxxxxxxxxxxxxxxxx		
x	nnn	xxxxxxx	nnn	hhmm	mmmm	hhmm	hhmm	xxxxxxxxxxxxxxxxxxxxxx		
x	nnn	xxxxxxx	nnn	hhmm	mmmm	hhmm	hhmm	xxxxxxxxxxxxxxxxxxxxxx		
x	nnn	xxxxxxx	nnn	hhmm	mmmm	hhmm	hhmm	xxxxxxxxxxxxxxxxxxxxxx		
x	nnn	xxxxxxx	nnn	hhmm	mmmm	hhmm	hhmm	xxxxxxxxxxxxxxxxxxxxxx		
x	nnn	xxxxxxx	nnn	hhmm	mmmm	hhmm	hhmm	xxxxxxxxxxxxxxxxxxxxxx		
x	nnn	xxxxxxx	nnn	hhmm	mmmm	hhmm	hhmm	xxxxxxxxxxxxxxxxxxxxxx		
x	nnn	xxxxxxx	nnn	hhmm	mmmm	hhmm	hhmm	xxxxxxxxxxxxxxxxxxxxxx		
x	nnn	xxxxxxx	nnn	hhmm	mmmm	hhmm	hhmm	xxxxxxxxxxxxxxxxxxxxxx		
x	nnn	xxxxxxx	nnn	hhmm	mmmm	hhmm	hhmm	xxxxxxxxxxxxxxxxxxxxxx		
x	nnn	xxxxxxx	nnn	hhmm	mmmm	hhmm	hhmm	xxxxxxxxxxxxxxxxxxxxxx		
x	nnn	xxxxxxx	nnn	hhmm	mmmm	hhmm	hhmm	xxxxxxxxxxxxxxxxxxxxxx		
x	nnn	xxxxxxx	nnn	hhmm	mmmm	hhmm	hhmm	xxxxxxxxxxxxxxxxxxxxxx		
x	nnn	xxxxxxx	nnn	hhmm	mmmm	hhmm	hhmm	xxxxxxxxxxxxxxxxxxxxxx		
x	nnn	xxxxxxx	nnn	hhmm	mmmm	hhmm	hhmm	xxxxxxxxxxxxxxxxxxxxxx		
x	nnn	xxxxxxx	nnn	hhmm	mmmm	hhmm	hhmm	xxxxxxxxxxxxxxxxxxxxxx		
x	nnn	xxxxxxx	nnn	hhmm	mmmm	hhmm	hhmm	xxxxxxxxxxxxxxxxxxxxxx		

OPTIONS: A=ADD,D=DELETE,U=UPDATE,*=PROCESSED,?=ERROR
 PROGRAM: SM75 MSG-INDX: nn -- DB.2.4 -- yy.ddd / hh:mm:ss
 MESSAGE:

To display, enter:

- 4 as the FUNCTION on the DB.2 Menu screen.
- 2.4 as the FUNCTION on the DB Menu screen.
- DB.2.4 as the FUNCTION on any other menu or formatted input screen.
- DB.2.4 or SCHD,JTRG as a top line command.

To exit:

- Press the PF3 key to return to the DB.2 Menu screen except with the TSO/ISPF interface when PF3 has been assigned to some other function.
- Move the cursor to the top line of the screen and enter the desired top line command if yet another function is desired.

3.2.9.1 Usage Notes

Use the JOB field to define the triggering job.

See 3.2.11.2, “Field Descriptions” on page 3-47 for other keyword descriptions.

3.2.10 DB.2.5 - Input Network Triggering Screen

Use this screen to define or review input networks which trigger job scheduling when their last workstation is logged out.

----- CA-7 INPUT NETWORK TRIGGERING -----

FUNCTION: xxxxxxxx (FORMAT,LIST,UPD)

PAGE nnnn

NWK: xxxxxxxx

DSNBR: nnnnnn

OPT	SCHID	TRGD-JOB	TRGID	DOTM	QTM	LDTM	SBTM	*---- EXCEPTIONS ----*
x	nnn	xxxxxxxx	nnn	hhmm	mmmm	hhmm	hhmm	xxxxxxxxxxxxxxxxxxxxxx
x	nnn	xxxxxxxx	nnn	hhmm	mmmm	hhmm	hhmm	xxxxxxxxxxxxxxxxxxxxxx
x	nnn	xxxxxxxx	nnn	hhmm	mmmm	hhmm	hhmm	xxxxxxxxxxxxxxxxxxxxxx
x	nnn	xxxxxxxx	nnn	hhmm	mmmm	hhmm	hhmm	xxxxxxxxxxxxxxxxxxxxxx
x	nnn	xxxxxxxx	nnn	hhmm	mmmm	hhmm	hhmm	xxxxxxxxxxxxxxxxxxxxxx
x	nnn	xxxxxxxx	nnn	hhmm	mmmm	hhmm	hhmm	xxxxxxxxxxxxxxxxxxxxxx
x	nnn	xxxxxxxx	nnn	hhmm	mmmm	hhmm	hhmm	xxxxxxxxxxxxxxxxxxxxxx
x	nnn	xxxxxxxx	nnn	hhmm	mmmm	hhmm	hhmm	xxxxxxxxxxxxxxxxxxxxxx
x	nnn	xxxxxxxx	nnn	hhmm	mmmm	hhmm	hhmm	xxxxxxxxxxxxxxxxxxxxxx
x	nnn	xxxxxxxx	nnn	hhmm	mmmm	hhmm	hhmm	xxxxxxxxxxxxxxxxxxxxxx
x	nnn	xxxxxxxx	nnn	hhmm	mmmm	hhmm	hhmm	xxxxxxxxxxxxxxxxxxxxxx
x	nnn	xxxxxxxx	nnn	hhmm	mmmm	hhmm	hhmm	xxxxxxxxxxxxxxxxxxxxxx
x	nnn	xxxxxxxx	nnn	hhmm	mmmm	hhmm	hhmm	xxxxxxxxxxxxxxxxxxxxxx
x	nnn	xxxxxxxx	nnn	hhmm	mmmm	hhmm	hhmm	xxxxxxxxxxxxxxxxxxxxxx
x	nnn	xxxxxxxx	nnn	hhmm	mmmm	hhmm	hhmm	xxxxxxxxxxxxxxxxxxxxxx
x	nnn	xxxxxxxx	nnn	hhmm	mmmm	hhmm	hhmm	xxxxxxxxxxxxxxxxxxxxxx
x	nnn	xxxxxxxx	nnn	hhmm	mmmm	hhmm	hhmm	xxxxxxxxxxxxxxxxxxxxxx

OPTIONS: A=ADD,D=DELETE,U=UPDATE,*=PROCESSED,?=ERROR

PROGRAM: SM75 MSG-INDX: nn -- DB.2.5 -- yy.ddd / hh:mm:ss

MESSAGE:

To display, enter:

- 5 as the FUNCTION on the DB.2 Menu screen.
- 2.5 as the FUNCTION on the DB Menu screen.
- DB.2.5 as the FUNCTION on any other menu or formatted input screen.
- DB.2.5 or SCHD,NTRG as a top line command.

To exit:

- Press the PF3 key to return to the DB.2 Menu screen except with the TSO/ISPF interface when PF3 has been assigned to some other function.
- Move the cursor to the top line of the screen and enter the desired top line command if yet another function is desired.

3.2.10.1 Usage Notes

Use the NWK and/or DSNBR keyword fields to define the triggering input network.

See 3.2.11.2, “Field Descriptions” on page 3-47 for other keyword descriptions.

3.2.11 DB.2.6 - Data Set Triggering Screen

Use this screen to define or review data sets which trigger job scheduling when either their creating jobs complete successfully or when the created/updated data set is closed. See also the POST AT CLOSE TIME field on the DB.6 screen and the 9.2.1.1, “Field Descriptions” on page 9-6.

CA-7 DATA SET TRIGGERING										
FUNCTION: xxxxxxxx (FORMAT,LIST,UPD)								PAGE nnnn		
DSN: xx								DSNBR: nnnnnn		
OPT	SCHID	TRGD-JOB	TRGID	DOTM	QTM	LDTM	SBTM	*----	EXCEPTIONS	----
x	nnn	xxxxxxxx	nnn	hhmm	mmmm	hhmm	hhmm	xxxxxxxxxxxxxxxxxxxxxxxx		
x	nnn	xxxxxxxx	nnn	hhmm	mmmm	hhmm	hhmm	xxxxxxxxxxxxxxxxxxxxxxxx		
x	nnn	xxxxxxxx	nnn	hhmm	mmmm	hhmm	hhmm	xxxxxxxxxxxxxxxxxxxxxxxx		
x	nnn	xxxxxxxx	nnn	hhmm	mmmm	hhmm	hhmm	xxxxxxxxxxxxxxxxxxxxxxxx		
x	nnn	xxxxxxxx	nnn	hhmm	mmmm	hhmm	hhmm	xxxxxxxxxxxxxxxxxxxxxxxx		
x	nnn	xxxxxxxx	nnn	hhmm	mmmm	hhmm	hhmm	xxxxxxxxxxxxxxxxxxxxxxxx		
x	nnn	xxxxxxxx	nnn	hhmm	mmmm	hhmm	hhmm	xxxxxxxxxxxxxxxxxxxxxxxx		
x	nnn	xxxxxxxx	nnn	hhmm	mmmm	hhmm	hhmm	xxxxxxxxxxxxxxxxxxxxxxxx		
x	nnn	xxxxxxxx	nnn	hhmm	mmmm	hhmm	hhmm	xxxxxxxxxxxxxxxxxxxxxxxx		
x	nnn	xxxxxxxx	nnn	hhmm	mmmm	hhmm	hhmm	xxxxxxxxxxxxxxxxxxxxxxxx		
x	nnn	xxxxxxxx	nnn	hhmm	mmmm	hhmm	hhmm	xxxxxxxxxxxxxxxxxxxxxxxx		
x	nnn	xxxxxxxx	nnn	hhmm	mmmm	hhmm	hhmm	xxxxxxxxxxxxxxxxxxxxxxxx		
x	nnn	xxxxxxxx	nnn	hhmm	mmmm	hhmm	hhmm	xxxxxxxxxxxxxxxxxxxxxxxx		
x	nnn	xxxxxxxx	nnn	hhmm	mmmm	hhmm	hhmm	xxxxxxxxxxxxxxxxxxxxxxxx		
x	nnn	xxxxxxxx	nnn	hhmm	mmmm	hhmm	hhmm	xxxxxxxxxxxxxxxxxxxxxxxx		
x	nnn	xxxxxxxx	nnn	hhmm	mmmm	hhmm	hhmm	xxxxxxxxxxxxxxxxxxxxxxxx		
x	nnn	xxxxxxxx	nnn	hhmm	mmmm	hhmm	hhmm	xxxxxxxxxxxxxxxxxxxxxxxx		
x	nnn	xxxxxxxx	nnn	hhmm	mmmm	hhmm	hhmm	xxxxxxxxxxxxxxxxxxxxxxxx		

OPTIONS: A=ADD,D=DELETE,U=UPDATE,*=PROCESSED,?=ERROR
 PROGRAM: SM75 MSG-INDX: nn -- DB.2.6 -- yy.ddd / hh:mm:ss
 MESSAGE:

To display, enter:

- 6 as the FUNCTION on the DB.2 Menu screen.
- 2.6 as the FUNCTION on the DB Menu screen.
- DB.2.6 as the FUNCTION on any other menu or formatted input screen.
- DB.2.6 or SCHD,DTRG as a top line command.

To exit:

- Press the PF3 key to return to the DB.2 Menu screen except with the TSO/ISPF interface when PF3 has been assigned to some other function.
- Move the cursor to the top line of the screen and enter the desired top line command if yet another function is desired.

3.2.11.1 Usage Notes

Use the DSN and/or DSNBR keywords to define the triggering data set.

See 3.2.11.2, “Field Descriptions” on page 3-47 for other keyword descriptions.

VSAM data sets and partitioned data sets (PDS) cannot be used as data set triggers.

3.2.11.2 Field Descriptions

FUNCTION:	The function to be performed. Value must be the name of some other screen or one of the following:
FORMAT	Clears the screen of user input data.
LIST	Lists the job names and schedule IDs for the triggering element.
UPD	Performs updating option indicated by OPT parameter below. Do a LIST function before an online UPD.
	Required: Yes
	Batch keyword: Positional parameter
n/a	This field has a constant value depending on the triggering element, either JTRG, DTRG, or NTRG for job, data set, or network respectively.
	Required: Yes for batch only
	Batch keyword: Positional parameter
JOB:	The job name whose successful completion causes triggering.
	Size/Type: 1 to 8 alphanumeric characters
	Required: Yes (for job triggers)
	Batch keyword: JOB
DSN:	The data set name whose creation/update causes triggering of a job.
	Size/Type: 1 to 44 alphanumeric characters
	Required: Yes (for data set triggers unless DSNBR is entered)
	Batch keyword: DSN
NWK:	The input network name whose completion causes triggering.
	Size/Type: 1 to 8 alphanumeric characters
	Required: Yes (for network triggers unless DSNBR is entered)
	Batch keyword: NWK
DSNBR:	The existing database member whose creation/completion causes triggering. It is the value assigned to the database member by CA-7.
	Size/Type: 1 to 6 numeric characters
	Required: Yes (for data set or network triggers unless DSN/NWK is entered)
	Batch keyword: DSNBR
OPT	Used with the UPD function to denote the operation to perform. Valid codes are A (add), D (delete), and U (update).
	Required: No
	Batch keyword: OPT

SCHID	<p>Indicates for which schedule ID of the triggering element the TRGD-JOB is to be scheduled. Is not valid with OPT=U.</p> <p>Size/Type: 1 to 3 numeric characters from 0 to 255</p> <p>Default: 0 (all schedule IDs)</p> <p>Required: No</p> <p>Batch keyword: SCHID</p>
TRGD-JOB	<p>The job name which is to be triggered by the completion/creation of the triggering element. Is not valid with OPT=U.</p> <p>Size/Type: 1 to 8 alphanumeric characters</p> <p>Required: Yes</p> <p>Batch keyword: TJOB</p>
TRGID	<p>Used to denote a replacement schedule ID to use when the TRGD-JOB is placed into the request queue.</p> <p>Size/Type: 1 to 3 numeric characters from 0 through 255</p> <p>Default: 0 (no replacement)</p> <p>Required: No</p> <p>Batch keyword: TRGID</p> <p>Note: If TRGID is used, it replaces the SCHID value of the triggered element. Any jobs triggered (by TRGD-JOB or data sets it creates) use this TRGID for their schedule ID unless they also have a TRGID value.</p> <p>Example:</p> <p>This field is useful to avoid a loop in a situation where a job is both triggered by and triggers the same job. For example, assume that JOBX is a backup job that needs to be run after the online system comes down. It triggers JOBY, an update job that further updates the database. JOBY then triggers JOBX again to get another backup. When you define the second occurrence of JOBX as being triggered by JOBY, use TRGID 2 so this occurrence of JOBX will not trigger JOBY again.</p>
DOTM	<p>Indicates the due-out time of day of TRGD-JOB rounded down to 10 minute increments. If this field is used, QTM must be omitted.</p> <p>Note: If used and the triggering job's DOTM is after the triggered job's DOTM, then CA-7 assumes the following calendar day.</p> <p>Size/Type: 4 numeric characters specified as hhmm, where hh can be 00 through 24 and mm can be 00 through 59. If specified, the lowest value is 10.</p> <p>Required: Only if QTM omitted</p> <p>Batch keyword: DOTM</p>

QTM

Indicates the elapsed queue time of TRGD-JOB rounded down to 10 minute increments. If this field is used, DOTM must be omitted since due-out time is then calculated as deadline time plus runtime, where deadline time is calculated as current date/time plus queue time.

Note: Either DOTM or QTM must be used.

Size/Type: 4 numeric characters specified as hhmm, where hh can be 00 through 24 and mm can be 00 through 59, the highest value being 2400

Required: Only if DOTM omitted

Batch keyword: QTM

LDTM

Indicates the elapsed lead time for TRGD-JOB rounded to 10 minute increments. This field specifies the lead or processing time necessary to ensure that TRGD-JOB meets its due-out time.

Size/Type: 4 numeric characters specified as hhmm, where hh can be 00 through 24 and mm can be 00 through 59

Default 0000

Required: No

Batch keyword: LEADTM

SBTM

A field which imposes a submit time of day requirement on TRGD-JOB. When used, the job is not submitted before this time. SBTM is always rounded down to 15 minute increments.

Note: If QTM is used, the date for the submit time requirement will be the same as the deadline start date. If DOTM is used and the SBTM is less than the DOTM, the date for the SBTM is the same as the deadline start date. Otherwise, the SBTM date is the previous day.

Size/Type: 4 numeric characters specified as hhmm, where hh can be 00 through 24 and mm can be 00 through 59

Default: 0 (no specific submit time requirement)

Required: No

Batch keyword: SBTM

EXCEPTIONS This is informational only. Brief messages appear here when an exception occurs trying to process the entry on this line.

Batch keyword: Not applicable

Messages which can appear and their meanings are as follows:

DOTM/QTm CONFLICT

Only one of these fields can be specified.

ERROR FROM: modulename

Internal logic error. Contact your installation's CA-7 specialist for assistance.

INVALID VALUE: fieldname

The value for the indicated field is not valid.

NO ROOM IN DATABASE

The CA-7 database is full. Contact your installation's CA-7 specialist for assistance.

SCHID 0 ALREADY EXISTS

Trigger is already defined for schedule ID zero which applies to all schedule IDs.

SECURITY EXCEPTION

You are not authorized to access that job.

TOO HIGH FOR: fieldname

The numeric value for the indicated field exceeds the maximum limit.

TOO LOW FOR: fieldname

The numeric value does not meet a valid range.

TRGD-JOB ACCESS ERROR

An I/O error occurred while reading the triggered job from the database.

TRGD-JOB NOT FOUND

The triggered job is not defined in the database.

TRIGGER ALREADY EXISTS

Trigger is already defined for this schedule ID.

WARNING: TRIGGER LOOP

Job is triggering itself. This is only a warning, and the database is updated.

3.2.11.3 Batch Input Example

This batch input example defines a trigger schedule from job CA07XX03 to job CA07XX05 with a queue time of 1 hour and a lead time of 30 minutes.

```
SCHD
TRGR
UPD,JTRG,OPT=A,JOB=CA07XX03,TJOB=CA07XX05,
LEADTM=0030,QTM=0100
DBM
```

3.2.12 DB.2.7 - Modification to Resolved Schedule Dates Screen

This screen provides the user with a rapid and direct method of permanently altering the resolved schedule information for either a job or input network without having to totally redefine the schedule. Such schedule modifications can be made any time after the schedule has been resolved and are effective immediately.

```

----- CA-7 MODIFICATION TO RESOLVED SCHEDULE DATES -----
FUNCTION: xxxxxxxx (FORMAT,LIST,UPD)

JOB: xxxxxxxx      NETWORK: xxxxxxxx      MODSTAT: xxxxxxxx
SCHID: nnn                                     YEAR: yyyy

      1      1      2      2      3 3
    ....5....0 ....5....0 ....5....0 1
JAN  nnnnnnnnnn nnnnnnnnnn nnnnnnnnnn n
FEB  nnnnnnnnnn nnnnnnnnnn nnnnnnnnnn
MAR  nnnnnnnnnn nnnnnnnnnn nnnnnnnnnn n
APR  nnnnnnnnnn nnnnnnnnnn nnnnnnnnnn
MAY  nnnnnnnnnn nnnnnnnnnn nnnnnnnnnn n
JUN  nnnnnnnnnn nnnnnnnnnn nnnnnnnnnn
JUL  nnnnnnnnnn nnnnnnnnnn nnnnnnnnnn n
AUG  nnnnnnnnnn nnnnnnnnnn nnnnnnnnnn n
SEP  nnnnnnnnnn nnnnnnnnnn nnnnnnnnnn
OCT  nnnnnnnnnn nnnnnnnnnn nnnnnnnnnn n
NOV  nnnnnnnnnn nnnnnnnnnn nnnnnnnnnn
DEC  nnnnnnnnnn nnnnnnnnnn nnnnnnnnnn n

PROGRAM: SM80 MSG-INDX: nn -- DB.2.7 -- yy.ddd / hh:mm:ss
MESSAGE:

```

To display, enter:

- 7 as the FUNCTION on the DB.2 Menu screen.
- 2.7 as the FUNCTION on the DB Menu screen.
- DB.2.7 as the FUNCTION on any other menu or formatted input screen.
- DB.2.7 or SCHDMOD as a top line command.

To exit:

- Press the PF3 key to return to the DB.2 Menu screen except with the TSO/ISPF interface when PF3 has been assigned to some other function.
- Move the cursor to the top line of the screen and enter the desired top line command if yet another function is desired.

3.2.12.1 Field Descriptions

There are no batch keywords since this is an online only function.

FUNCTION: The function to be performed. Value must be the name of some other screen or one of the following:

FORMAT Clears the screen of user input data.

LIST Lists schedule information from the database.

UPD Updates existing schedule information in the database using the screen values for processing days.

Required: Yes

JOB: The job name for which schedule information is to be listed or altered. If this field is used, NETWORK must be omitted.

Size/Type: 1 to 8 alphanumeric characters

Required: No

NETWORK: The name of an input network for which schedule information is to be listed or altered. If this field is used, JOB must be omitted.

Size/Type: 1 to 8 alphanumeric characters

Required: No

MODSTAT: Indicates the status of modifications made to the schedule. Values are:

blanks Indicates that no changes have been made to this schedule since the last RESOLV was done.

CURRENT Indicates that at least one modification to the schedule ID shown had been made previously with this screen and is still in effect.

OVERLAID Indicates that at least one modification had been made previously with this screen but a subsequent RESOLV for the schedule overlaid the modification that had been made without changing the schedule.

Default: Blanks

Required: No

SCHID: Indicates which schedule ID of the job or network is to be listed or updated.

Size/Type: 1 to 3 numeric characters

Required: Yes

YEAR:	Indicates the calendar year for the first month of the schedule displayed.
JAN thru DEC	<p>These values indicate the run and nonrun Julian days of the job or network for the month indicated. Any scheduled day (1) may be changed to a nonscheduled day (0), or vice versa, with the UPD function. This can be easily accomplished with a LIST followed by a UPD. Schedule Resolution negates these changes.</p> <p>1 Days on which work is done, a scheduled day.</p> <p>0 Days on which work is not done, a nonscheduled day.</p> <p>Note: If the schedule ID being reviewed was resolved on a July through June basis instead of a January through December basis, the month titles show JUL through DEC followed by JAN through JUN.</p> <p>Required: No</p>

3.2.12.2 Usage Notes

Unlike other DB functions, DB.2.7 is only available as an online function.

Once displayed, you may change the values as necessary to accomplish the desired changes.

Note: If schedule scan has already brought the job into the queue and the schedule has been adjusted for the day, then the DB.2.7 changes do not take effect for that day.

3.2.13 DB.2.8 - Base Calendar Maintenance

This screen provides the means to list, add, update and delete CA-7 base calendars. New and updated calendars are immediately available to other CA-7 functions, such as Resolve, without the need for a SCAL statement in the CA-7 initialization file.

To use online base calendar maintenance, the CA-7 calendar PDS must be defined to CA-7 through the CALENDAR statement in the CA-7 initialization file (see the *CA-7 Systems Programmer Guide*).

```

----- CA-7 BASE CALENDAR MAINTENANCE -----
FUNCTION: xxxxxxxx (FORMAT,LIST,ADD,UPD,DELETE,REFRESH)
CALENDAR: xxxxxxxx SCHONLY: x YEAR: nn


      1       1       2       2       3
    ....5....0....5....0....5....01
JAN  nnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnn   01 / 01   01 / 31
FEB  nnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnn   02 / 01   02 / 28
MAR  nnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnn   03 / 01   03 / 31
APR  nnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnn   04 / 01   04 / 30
MAY  nnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnn   05 / 01   05 / 31
JUN  nnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnn   06 / 01   06 / 30
JUL  nnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnn   07 / 01   07 / 31
AUG  nnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnn   08 / 01   08 / 31
SEP  nnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnn   09 / 01   09 / 30
OCT  nnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnn   10 / 01   10 / 31
NOV  nnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnn   11 / 01   11 / 30
DEC  nnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnn   12 / 01   12 / 31


PROGRAM: SM85 MSG-INDX: nn -- DB.2.8 -- yy.ddd / hh:mm:ss
MESSAGE:
```

To display, enter:

- 8 as the FUNCTION on the DB.2 Menu screen.
- 2.8 as the FUNCTION on the DB Menu screen.
- DB.2.8 as the FUNCTION on any other menu or formatted input screen.
- DB.2.8 or CALMOD as a top line command.

To exit:

- Press the PF3 key to return to the DB.2 Menu screen except with the TSO/ISPF interface when PF3 has been assigned to some other function.
- Move the cursor to the top line of the screen and enter the desired top line command if yet another function is desired.

3.2.13.1 Field Descriptions

There are no batch keywords since this is an online only function.

FUNCTION: The function to be performed. Value must be the name of some other screen or one of the following:

FORMAT Clears the screen of user input data.

LIST Lists data for the calendar specified in the CALENDAR field.

ADD Adds the calendar displayed. The YEAR field must be specified to add a new calendar.

UPD Updates the calendar displayed. Any changes made to the calendar take effect immediately. That is, if you update a calendar and then do a RESOLVE against a job using that calendar, the updated version of the calendar will be used.

DELETE Deletes the calendar specified in the CALENDAR field. The copy of the calendar saved in the CA-7 calendar PDS is deleted. The delete function has no effect on calendars which reside in load libraries.

REFRESH Fetches the load module copy of the calendar specified in the CALENDAR field and uses it to replace the copy in the CA-7 calendar PDS.

Required: Yes

CALENDAR: Name of the CA-7 calendar you wish to act on. For a calendar to be used by CA-7 to resolve job/network schedules the name must be in the format SCALyyxx; where, yy is the last 2 digits of the year it represents and xx is a two-position suffix to uniquely identify the calendar. All calendar names that begin with SCAL must adhere to this format. However, you may create and save calendars under other names to use as models or templates for the creation of other calendars.

Size/Type: 1 to 8 alphanumeric characters

Required: Yes

SCHONLY: If Y (yes) is specified, then CA-7 will count only available processing days when resolving a schedule with number of days relative to the beginning or end of the month (see RDAY field on the 3.2.2, “DB.2.1 - CPU Job Scheduling Screen” on page 3-11). If SCHONLY is set to N (no), CA-7 will count all days when calculating days relative to the beginning or end of the month.

Default: N (NO)

Required: No

YEAR: Specifies the year that the calendar represents. When using the standard CA-7 calendar naming convention (SCALyyxx), the YEAR should match positions 5 and 6 of the calendar name.

Calendars can be defined for the twentieth or twenty-first centuries. Years 75 through 99 are considered to be part of the twentieth century (that is, 98 will be interpreted as 1998). Years 00 through 74 are considered to be part of the twenty-first century (that is, 02 will be interpreted as 2002).

Size/Type: 2 numeric characters

Required: Yes - for ADD or UPD functions if the calendar name is not in standard SCALyyxx format

DAYS OF THE MONTH:

Each month is represented by a string of zeros and ones. Each digit represents a particular day of that month. If the digit for a day is 1 (one), it indicates that it is a 'processing day'. If the digit is 0 (zero), it indicates that it is not a processing day. The number of digits in an individual string should represent the number of days in that calendar month.

Required: Yes - for ADD or UPD functions

BEGIN and END:

For each month you may specify the beginning and ending days of that month. It must be used if the calendar being defined has non-standard months; for example, a fiscal or accounting calendar. A year-end boundary may only be crossed in the first or twelfth month.

Required: Yes - for ADD or UPD functions

3.2.13.2 Usage Notes

1. Unlike other DB functions, DB.2.8 is only available as an online function.
2. To use online base calendar maintenance the CA-7 calendar PDS must be defined to CA-7 through the CALENDAR statement in the CA-7 initialization file (see the *CA-7 Systems Programmer Guide*).
3. Base calendars will be automatically added to the CA-7 calendar PDS when:
 - a. A CALBLK statement is included in the CA-7 initialization file and the load module copy exists but there is no copy in the PDS.
 - b. A reference is made to the calendar from the online base calendar maintenance facility or with a RESOLVE or PRINT topline command and the load module copy exists but there is no copy in the PDS.
 - c. An ADD function is performed from the online base calendar maintenance facility.
4. If you have assembled and link edited a new copy of a calendar and want to propagate it to the CA-7 calendar PDS, use the REFRESH function. This will replace the existing copy in the CA-7 calendar PDS with a new converted copy of the load library version.
5. Base calendars which reside in the CA-7 calendar PDS do not require a CALBLK statement in the CA-7 initialization file.
6. Calendars with nonstandard names can be created using the online base calendar maintenance facility and used as patterns or models to create standard base calendars. All calendars stored in the CA-7 calendar PDS must have valid PDS member names (1-8 characters starting with an alpha or national character).
7. The DELETE function removes the specified calendar from the CA-7 calendar PDS ONLY. It has no effect on a load module copy of the same calendar.
8. To list all calendar names in the CA-7 calendar PDS, issue the following topline command:

```
LISTDIR,DSN=*SCAL*
```

Chapter 4. Requirement Definitions

This chapter describes how CA-7 determines requirements from JCL, how to define and maintain job requirements, and how requirements are satisfied.

4.1 Execution Requirements and the LOAD Process

4.1.1 Dependence Definition

As work is defined in the CA-7 database, dependencies (preexecution or predecessor requirements) can be established so that work is not initiated until these dependencies are satisfied.

Any job under the control of CA-7 can have predecessor requirements. CA-7 does not submit the job until all requirements are satisfied even though time scheduling criteria has been met.

The main categories of predecessor requirements are as follows:

- Successful completion of predecessor jobs
- Availability of data sets
- Date/time window
- Completion of an input network
- Mutual exclusivity with other jobs
- Specified manual activities such as JCL overrides and manual verifications
- Presence of other jobs in the CA-7 queues

Note: Jobs should not have more than 255 requirements.

4.1.2 Database Definitions

This chapter discusses predecessor requirement database definitions for CPU job, input network, user requirement, and data set predecessors.

It also discusses how to define those output networks which must be invoked upon normal completion of a job and how to define report IDs which may be produced by a job.

4.1.3 Data Set Dependencies

The LOAD process addresses the requirements associated with data set availability. Based on JCL, the LOAD process determines if the job being LOAded uses or creates a particular data set and updates the database accordingly. Data set dependencies are established for the job for each data set that is input to the job. Unless the dependent job is marked as a MAINT job on the DB.1 screen, it does not run before any required data set is created.

Whenever there is a permanent change to the JCL, the job must go back through the LOAD process which redefines the data set requirements for the job. LOAD is automatic if the JCL is changed through the DB.7 screen.

The LOAD process neither determines nor changes requirements except those associated with data set availability. If the job has requirements other than those associated with data set availability that change, they must be specified through the various DB.3 screen functions. The LOAD process does not delete any requirements that were manually updated or added through a DB.3 screen. This means that data set name changes require DB.3 updates to delete outdated references unless RENAME was used on the DB.6 screen before LOAD occurred.

4.1.4 Other Dependencies

The DB.1 screen contains fields such as HOLD, VERIFY, and JCL-OVRD with which some predecessor requirements can be permanently defined for a job. Top line commands HOLD, VERIFY, JCLOVRD, and ADDRQ are available for dynamically defining predecessor requirements on a temporary basis. The QM.1 screen allows for dynamically adding some of these same predecessor requirements to jobs on a temporary basis.

A job's JCL can also include, on a scheduled basis if desired, commands to cause some of the same predecessor requirements to be applied to the job when the job is scheduled into the queues. This is accomplished with a special set of commands known as Scheduled Overrides which are discussed further on 10.2, "Scheduled Overrides" on page 10-6. Based on how these are scheduled, they may be permanent or temporary without having been defined to the database.

4.1.5 Temporary Predecessors

Predecessor requirements are normally defined on a permanent basis. However, on occasion it may become necessary on a onetime basis to temporarily add new ones or skip existing ones.

The DB.3 screens provide the ability to do that with the NEXT-RUN facility. Any such onetime definitions are automatically deleted from the database after a single execution of the dependent job.

A job executing with any such temporary predecessors, either added or skipped, is identified on LQ job inquiry output as executing with "temporary requirements." LJOB,LIST=RQMT, and so forth, also flag individual predecessors as SKIP or ONLY to indicate when these settings exist. QM.2 update screens reflect the letter O to identify a NEXT-RUN=ONLY predecessor while the letter A identifies one added through the ADDRQ top line command. NEXT-RUN=SKIP predecessors do not appear on the QM.2 update screens as they are skipped when the job enters the queue.

4.2 Satisfying Requirements

The system examines all of a job's defined predecessor requirements each time the job is scheduled. It itemizes these requirements and establishes a total count of the unsatisfied requirements for the job when it enters the request queue. As each unsatisfied requirement is fulfilled, the count of the unsatisfied requirements is decremented. This count is known as the master count and appears on many screens as MCNT. When the count reaches zero, the job moves to the ready queue and then is available for execution.

Outstanding requirements can be satisfied either manually or automatically. The requirements automatically tracked and satisfied by the system are as follows:

- Successful job completion of a predecessor job under CA-7's control
- Input network completion of a network connection

When the last workstation in an input network is logged complete by the workstation operator, CA-7 posts this network requirement as complete for any dependent jobs.

- Data set availability posting by:
 - SMF feedback
 - Batch card load program (BLCP)
 - Trailer step
 - U7SVC
- System date/time specifications

Most outstanding requirements needing manual intervention are related to:

- user requirements
- manually prepared overrides
- manual verifications
- restart requirements
- hold requirements

These requirements must be posted by the POST command, QM.1 type screens, or other top line commands such as:

SUBTM

VERIFY

The QM.2 screen can be used to view the requirements of a job in the request queue. The user can see both the outstanding and satisfied requirements for a job. The user can also post and unpost requirements for a job by using this screen.

If requirements for a job are predictable, they may lend themselves to use of special trailer step job steps. See Chapter 2 of the *CA-7 Interfaces Guide*.

4.3 Defining Predecessors and Successors

4.3.1 DB.3 - Job Predecessor/Successor Menu Screen

Use the DB.3 Menu screen to select various screens for defining "connections" (predecessor requirements) to CPU jobs.

```

----- CA-7 JOB PREDECESSOR/SUCCESSOR MENU -----
FUNCTION ==> xxxxxxxx

EXECUTION REQUIREMENTS DEFINED BY:
  1 - DATA SET PREDECESSORS
  2 - CPU JOB PREDECESSORS AND
      MUTUALLY EXCLUSIVE JOBS (CAN NOT RUN AT SAME TIME)
  4 - INPUT NETWORK PREDECESSORS AND
      OUTPUT NETWORK SUCCESSORS
  6 - USER MEMO-FORM PREDECESSORS
  7 - REPORT IDS CREATED

PROGRAM: SM60 MSG-INDX: nn  -- DB.3      -- yy.ddd / hh:mm:ss
MESSAGE:

```

To display, enter:

- 3 as the FUNCTION on the DB Menu screen.
- DB.3 as the FUNCTION value on any other menu or formatted input screen.
- DB.3 or JOBCONN as a top line command.

To exit:

- Press the PF3 key to return to the DB Menu screen except with the TSO-ISPF interface when PF3 has been assigned to some other function.
- If some other function is desired, enter the name of an online screen as the FUNCTION value or move the cursor to the top line and enter a top line command.

4.3.1.1 Usage Notes

Select the desired function by entering the appropriate FUNCTION value and pressing the Enter key.

4.3.2 DB.3.1 - Data Set Predecessors Screen

Data set requirements are initially set by CA-7 at initial job LOAD time. In other words, when a job runs for the first time, CA-7 automatically connects the job's input data sets as requirements when the job completes. The top line LOAD command can also be issued to accomplish this. The DB.3.1 screen allows the user to modify such requirements or to add data set requirements for data sets that are not used as input for the job.

[illegible]

To display, enter:

- 1 as the FUNCTION on the DB.3 Menu screen.
- 3.1 as the FUNCTION on the DB Menu screen.
- DB.3.1 as the FUNCTION value on any other menu or formatted input screen.
- DB.3.1 or JOBCONN,DSN as a top line command.

To exit:

- Press the PF3 key to return to the DB.3 Menu screen except with the TSO-ISPF interface when PF3 has been assigned to some other function.
- Move the cursor to the top line of the screen and enter the desired top line command if yet another function is desired.

4.3.2.1 Field Descriptions

FUNCTION:	The function to be performed. Value must be the name of some other screen or one of the following: FORMAT Clears the screen of user input data. LIST Lists data from the database. In batch, a formatted screen is not listed; only a found or not found message is returned. UPD Updates a data set requirement according to OPT field values. Required: Yes Batch keyword: Positional parameter
n/a	Predecessor type. Has a constant value of DSN for data sets. Required: Yes for batch only Batch keyword: Positional parameter
PRED FOR JOB:	The required job name for which data set predecessors are being defined. Size/Type: 1 to 8 alphanumeric characters Required: Yes Batch keyword: Positional parameter
LIST-SCHID:	Applies only to the LIST function. When used, all connections for the SCHID entered and all connections for schedule ID of 0 are listed. Schedule IDs on each detail line apply to that line only. Size/Type: 1 to 3 numeric characters Default: Null (Causes connections for all schedule IDs to be listed) Required: No Batch keyword: Not applicable
OPT	This field is used with the UPD function to denote the type of connection operation. The operation codes are A (add), D (delete), and U (update). Note: The U option applies only to LEADTM, PERM, and NEXT-RUN fields. Required: No Batch keyword: OPT

SCHID

Indicates for which schedule ID of this job the data set requirement is applied. The user cannot specify the zero default for one connection and use a nonzero schedule ID for another connection to the same job and data set. An attempt to make such a connection results in an error message issued by CA-7. (For an online LIST function, the SCHID field on the same line as JOB field defaults to null, causing all schedule IDs to be listed.) This field is not valid with OPT=U.

Size/Type: 1 to 3 numeric characters from 0 through 255

Default: 0 (requirement holds for all schedule IDs)

Required: No

Batch keyword: SCHID

LEADTM

Represents satisfaction lead time in hours. If nonzero, this value overrides any satisfaction lead time indicated on the DB.1 screen for data sets. Values are:

- 0** The data set must have been created or updated since the start time of the last run of the job. There is no special consideration for the amount of time which may have elapsed.
- 99** The requirement is never considered initially satisfied when the job enters the request queue. The data set must be created or updated while this job is in the request queue.
- nn** The data set must have been created or updated within the last nn hours. Values for nn may be 1 to 98.

Default: 0

Required: No

Batch keyword: LEADTM

DATASET NAME

The fully qualified data set name which is to be defined as a data set requirement for the job in the JOB field. Is not valid with OPT=U.

Size/Type: 1 to 44 alphanumeric characters

Required: Yes (unless DSNBR is used)

Batch keyword: DSN

DSNBR	<p>The data set number (as previously assigned by CA-7) which is to be defined as a data set requirement for the job in the JOB field. Is not valid with OPT=U.</p> <p>Size/Type: 1 to 6 numeric characters</p> <p>Required: Yes (unless DSNAME is used)</p> <p>Batch keyword: DSNBR</p>
PERM	<p>Indicates whether this data set is to be considered permanent by CA-7 for this job's requirements (Y or N). The DB.6 screen may be used to mark a data set as permanent for all jobs. This field defines the data set as permanent only for this job.</p> <p>Default: N</p> <p>Required: No</p> <p>Batch keyword: PERM</p>
NEXT-RUN	<p>Indicates status of this predecessor for the next run of this job. The next time the job is scheduled into the queues, the predecessors for that execution are determined by this value.</p> <p>YES Indicates normal, recurring predecessor.</p> <p>ONLY Indicates onetime predecessor for only the next run of this job. Automatically discarded by CA-7 when next successful job completion is done. (See note.) Only valid when OPT=A. Noted on QM.2 display as O and on various general inquiry displays as ONLY.</p> <p>SKIP Indicates normal, recurring predecessor which is to be skipped (ignored) when the next run is scheduled into the queues. Only valid when OPT=U and previous value was YES. Automatically reinstated to YES when next successful job completion is done. (See note.) Noted on certain displays as SKIP but does not appear on QM.2 displays since it is skipped when the job enters the queues.</p> <p>Default: Yes (for OPT value of A)</p> <p>Required: No</p> <p>Batch keyword: NEXTRUN</p> <p>Note: The "next successful job completion" only applies to jobs that are not already in the queue when the NEXT-RUN is set. However, if the job is already in the queue and entered the queue when the NEXT-RUN was SKIP or ONLY, then it is reset as indicated.</p>

4.3.2.2 Usage Notes

You may add additional requirements on a onetime basis. You may skip existing requirements on a onetime basis.

A data set connection is not considered a true predecessor requirement if any one of the following conditions are true:

- The data set is TYPE=PERM on DB.6 screen.
- The requirement is marked PERM for this job on the DB.3.1 screen.
- The job is marked MAINT=Y on the DB.1 screen.
- The requirement does not apply to this run (schedule ID dependency).
- The requirement was defined as NEXT-RUN=SKIP.
- The job is scheduled by a RUN or LOAD command.
- The job was DEMANDED using TYPE=RES.

The combination of job name, job schedule ID, and data set name handles any unique variations of the data set connection.

If the LOADDNS keyword is used on the DBASE statement in the initialization file, then the LOAD process does not build any DD or data set information for jobs that are marked MAINT=Y on the DB.1 (JOB) screen. This means that there will not be any data set connections for these jobs unless added manually.

4.3.2.3 Batch Input Example

This batch input example adds the data set USER.XX00.WEEKLY as a predecessor requirement to job CA07XX01. This requirement applies to the next run scheduled and then is removed.

```
JOBCONN
UPD,DSN,CA07XX01,OPT=A,DSN=USER.XX00.WEEKLY,NEXTRUN=ONLY
DBM
```

4.3.3 DB.3.2 - CPU Job Predecessors Screen

Use this screen to establish predecessor job requirements.

[illegible]

To display, enter:

- 2 as the FUNCTION on the DB.3 Menu screen.
- 3.2 as the FUNCTION on the DB Menu screen.
- DB.3.2 as the FUNCTION value on any other menu or formatted input screen.
- DB.3.2 or JOBCONN,JOB as a top line command.

To exit:

- Press the PF3 key to return to the DB.3 Menu screen except with the TSO-ISPF interface when PF3 has been assigned to some other function.
- Move the cursor to the top line of the screen and enter the desired top line command if yet another function is desired.

4.3.3.1 Field Descriptions

FUNCTION:	The function to be performed. The value must be the name of some other screen or one of the following:
FORMAT	Clears the screen of user input data.
LIST	Lists data from the database. In batch, a formatted screen is not listed; only a found or not found message is returned.
UPD	Updates predecessor job requirements in the database according to OPT field values.
	Required: Yes
	Batch keyword: Positional parameter
n/a	Predecessor type. Has a constant value of JOB for jobs.
	Required: Yes for batch only
	Batch keyword: Positional parameter
PRED FOR JOB:	The job name for which job requirements are being defined or listed.
	Size/Type: 1 to 8 alphanumeric characters
	Required: Yes
	Batch keyword: Positional parameter
LIST-SCHID	Applies only to the LIST function. A SCHID of 0 applies to connections for all schedules and therefore is listed with connections for any schedule ID requested. Schedule IDs on each detail line apply to that line only.
	Size/Type: 1 to 3 numeric characters
	Default: Null (all connections for all schedule IDs appear)
	Required: No
	Batch keyword: Not applicable
OPT	This field is used with the UPD function to denote the type of connection operation. The operation codes are A (add), D (delete), and U (update).
	Note: The U option applies only to the LEADTM and NEXT-RUN fields.
	Required: No
	Batch keyword: OPT

SCHID

Indicates for which schedule ID (of this job, not the PRED-JOB) the requirement is applied. If omitted when the connection is made, default is 0. A zero default cannot be specified for one connection and a nonzero schedule ID used for another connection to the same job with the same predecessor job. An attempt to make such a connection results in an error message issued by CA-7. (For the online LIST function, the field on the same line as JOB field defaults to null causing all schedule IDs to be listed.)

Size/Type: 1 to 3 numeric characters from 0 through 255
 Default: 0 (indicates the requirement holds for all schedule IDs)
 Required: No
 Batch keyword: SCHID

LEADTM

Indicates satisfaction lead time in hours. If nonzero, this value overrides any satisfaction lead time indicated on the DB.1 screen for this job connection. Values are:

- 0** The predecessor job must have started and completed successfully since the last run of the dependent job. There is no special consideration for the amount of time which may have elapsed.
- 99** The requirement is never considered initially satisfied when this job enters the request queue. The dependent job must complete normally while this job is in the request queue.
- nn** Since the last run of this job, the predecessor job has started and completed successfully within the last nn hours. Values for nn may be 1 to 98.

Default: 0
 Required: No
 Batch keyword: LEADTM

PRED-JOB

Names the job on which the job specified in the JOB field is dependent. This predecessor job name may be preceded by a slash (/) to indicate a negative job dependency. A conditional job dependency is indicated by prefixing the predecessor job name with a question mark (?). If job A is conditionally dependent on job B, then job A will depend on job B only if job B is in the request, ready, or active queues when job A enters the request queue.

If a generic negative dependency is specified, the successor job will only be submitted when there are not any jobs submitted that meet the generic PRED-JOB criteria. If a generic dependency is defined without the slash, the successor job will enter the request queue with the generic requirement. The first job that completes and meets the generic PRED-JOB will satisfy that requirement. (CA-7 produces a warning message indicating that PRED-JOB is not found.)

Size/Type: 1 to 9 alphanumeric characters
(or 1 to 8 alphanumeric characters terminated
with an asterisk to indicate a generic name)

Required: Yes

Batch keyword: PRED

Note: A job predecessor requirement defined as conditional (?) is considered satisfied if it is NOT in the request, ready, or active queue when the requiring job enters the request queue. Otherwise, the predecessor job is considered an unsatisfied requirement. Also, a conditional requirement can establish only one requirement per job even if there are multiple occurrences in the queues for the other job.

A negative job dependency reflects mutual exclusion of jobs and prevents them from executing at the same time. To guarantee exclusion, a similar connection must also be made to define this job as mutually exclusive for the predecessor job. Lead time is not valid with negative job dependencies.

Negative job dependencies are checked when a job is in the ready queue preparing for submission (all other job requirements have been satisfied). If one of the job's mutually exclusive jobs has been submitted (ready queue or active queue), then a positive job dependency is assigned to the job and it is moved back to the request queue. The job will then wait for the successful completion of the mutually exclusive job before it will be ready for submission. A job requirement is assigned even if the active job was requested with a RUN function or command. The completion of the active job from RUN does not POST the requirement; the requirement must be posted manually.

NEXT-RUN

Indicates status of this predecessor for the next run of this job. The next time the job is scheduled into the queues, the predecessors for that execution are determined by the following values:

- YES** Indicates normal, recurring predecessor.
- ONLY** Indicates onetime predecessor for only the next run of this job. Automatically discarded by CA-7 when next successful job completion is done. (See note.) Noted on QM.2 display as O and on various general inquiry displays as ONLY.
- SKIP** Indicates normal, recurring predecessor which is to be skipped (ignored) when the next run is scheduled into the queues. Only valid when OPT=U and previous value was YES. Automatically reinstated to YES when next successful job completion is done. (See note.) Noted on certain displays as SKIP but does not appear on QM.2 displays since it is skipped when the job enters the queues.

Default: Yes (for OPT value of A)

Required: No

Batch keyword: NEXTRUN

Note: The "next successful job completion" only applies to jobs that are not already in the queue when the NEXT-RUN is set. However, if the job is already in the queue and entered the queue when the NEXT-RUN was SKIP or ONLY, then it is reset as indicated.

4.3.3.2 Usage Notes

You may add additional requirements on a onetime basis. You may skip existing requirements on a onetime basis.

A job connection is not considered a true requirement for the following conditions:

- The job is scheduled by a RUN or LOAD command.
- The requirement does not apply to this run (schedule ID dependency).
- The requirement was defined as NEXT-RUN=SKIP.
- The job was DEMANDED using TYPE=RES.

The combination of the job name, job schedule ID, and predecessor job name handles the unique variations of the job connection.

4.3.3.3 Batch Input Example

This batch input example adds job CA07XX03 as a predecessor requirement for job CA07XX05.

```
JOBCONN  
UPD,JOB,CA07XX05,OPT=A,PRED=CA07XX03  
DBM
```


4.3.4 DB.3.4 - Input/Output Network Tasks Screen

Use this screen to define a CPU job's requirements (or connections) for workstation networks. One network can be connected to multiple CPU jobs.

CA-7 INPUT/OUTPUT NETWORK TASKS							
FUNCTION: xxxxxxxx (FORMAT,LIST,UPD)							PAGE nnnn
TASK FOR JOB: xxxxxxxx				LIST-SCHID: nnn			
OPT	SCHID	LEADTM	NETWORK	SUB-ID	NWK-SCHID	DESCRIPTION	NEXT-RUN
x	nnn	nnnn	xxxxxxx	xxxxxxx	nnn	xxxxxxx	xxxx
x	nnn	nnnn	xxxxxxx	xxxxxxx	nnn	xxxxxxx	xxxx
x	nnn	nnnn	xxxxxxx	xxxxxxx	nnn	xxxxxxx	xxxx
x	nnn	nnnn	xxxxxxx	xxxxxxx	nnn	xxxxxxx	xxxx
x	nnn	nnnn	xxxxxxx	xxxxxxx	nnn	xxxxxxx	xxxx
x	nnn	nnnn	xxxxxxx	xxxxxxx	nnn	xxxxxxx	xxxx
x	nnn	nnnn	xxxxxxx	xxxxxxx	nnn	xxxxxxx	xxxx
x	nnn	nnnn	xxxxxxx	xxxxxxx	nnn	xxxxxxx	xxxx
x	nnn	nnnn	xxxxxxx	xxxxxxx	nnn	xxxxxxx	xxxx
x	nnn	nnnn	xxxxxxx	xxxxxxx	nnn	xxxxxxx	xxxx
x	nnn	nnnn	xxxxxxx	xxxxxxx	nnn	xxxxxxx	xxxx
x	nnn	nnnn	xxxxxxx	xxxxxxx	nnn	xxxxxxx	xxxx
x	nnn	nnnn	xxxxxxx	xxxxxxx	nnn	xxxxxxx	xxxx
x	nnn	nnnn	xxxxxxx	xxxxxxx	nnn	xxxxxxx	xxxx
OPTIONS: A=ADD,D=DELETE,U=UPDATE,*=PROCESSED,?=ERROR							
PROGRAM: SM61 MSG-INDX: nn -- DB.3.4 -- yy.ddd / hh:mm:ss							
MESSAGE:							

To display, enter:

- 4 as the FUNCTION on the DB.3 Menu screen.
- 3.4 as the FUNCTION on the DB Menu screen.
- DB.3.4 as the FUNCTION on any other menu or formatted input screen.
- DB.3.4 or JOBCONN,NWK as a top line command.

To exit:

- Press the PF3 key to return to the DB.3 Menu screen except with the TSO-ISPF interface when PF3 has been assigned to some other function.
- Move the cursor to the top line of the screen and enter the desired top line command if yet another function is desired.

4.3.4.1 Field Descriptions

FUNCTION:	The function to be performed. Value must be the name of some other screen or one of following:
FORMAT	Clears the screen of user input data.
LIST	Lists data from the database. In batch, a formatted screen is not listed; only a found or not found message is returned.
UPD	Updates a network requirement according to OPT field values.
	Required: Yes
	Batch keyword: Positional parameter
n/a	Predecessor type. Has a constant value of NWK for networks.
	Required: Yes for batch only
	Batch keyword: Positional parameter
TASK FOR JOB:	The job name for which a network connection is to be established.
	Size/Type: 1 to 8 alphanumeric characters
	Required: Yes
	Batch keyword: Positional parameter
LIST-SCHID:	Applies only to the LIST function. A SCHID of 0 applies to connections for all schedules and therefore is listed with connections for any schedule ID requested. Schedule IDs on each detail line apply to that line only.
	Size/Type: 1 to 3 numeric characters
	Default: Null (causes connections for all schedule IDs to appear)
	Required: No
	Batch keyword: Not applicable
OPT	This field is used with the UPD function operation. The operation codes are A (add), D (delete), and U (update).
	Note: The U option applies only to the LEADTM, NWKSCHID, DESCRIPTION, and NEXT-RUN fields.
	Required: No
	Batch keyword: OPT

SCHID

Indicates the schedule ID (of this job) for which a network requirement is applied. If omitted when the connection is made, the default is 0. A zero default cannot be specified for one connection and a nonzero schedule ID for another connection to the same job with the same network and sub-ID. An attempt to make such a connection results in an error message issued by CA-7.

Size/Type: 1 to 3 numeric characters from 0 through 255

Default: 0 (indicates that the requirement holds for all schedule IDs)

Required: No

Batch keyword: SCHID

LEADTM

For an input network, satisfaction lead time in hours. Values are:

0 The requirement is satisfied if the input network has been processed since the last run of the job.

9999 The requirement is never considered initially satisfied when the job enters the request queue. The preprocess network must complete while this job is in the request queue.

nnnn The requirement is only considered satisfied if the network has been completed within the last nnnn hours (0 to 9998).

Default: 0

Required: No

Batch keyword: LEADTM

For an output network, all postprocess network due-out times are adjusted (lead time is added) by this value.

Size/Type: 1 to 4 numeric characters specified as hhmm where hours are 00 through 24 and minutes are 00 through 59

Default: 0

Required: No

Batch keyword: LEADTM

NETWORK

The network name.

Size/Type: 1 to 8 alphanumeric characters

Required: Yes

Batch keyword: NETWORK

SUB-ID

A name which further qualifies the network. For example, if a network is connected to a job multiple times with the same SCHID, the user must further qualify the network name, by specifying a SUB-ID, to distinguish between the various purposes each network connection fulfills. For output networks only.

Size/Type: 1 to 8 alphanumeric characters

Required: No

Batch keyword: SUBID

NWK-SCHID

A network schedule identification. This field indicates which output network schedule ID is to be used when the job is scheduled. This field has no effect for input networks. If this field does not match the SCHID field on the DB.2.3 screen, the output network is not scheduled.

Size/Type: 1 to 3 numeric characters specified as 0 through 255

Default: 0 (the job's schedule ID is to be used)

Required: No

Batch keyword: NWKSCHID

DESCRIPTION

Further identifies the connected network.

For output networks, this field shows when listing the network.

For input networks, this field is only documentation.

Size/Type: 1 to 8 alphanumeric characters

Required: No

Batch keyword: DESC

NEXT-RUN

Indicates status of this predecessor for the next run of this job. The next time the job is scheduled into the queues, the predecessors for that execution are determined by the following:

YES Indicates normal, recurring predecessor.

ONLY Indicates onetime predecessor for only the next run of this job. Automatically discarded by CA-7 when next successful job completion is done. (See note.) Only valid when OPT=A. Noted on QM.2 display as O and on various general inquiry displays as ONLY.

SKIP Indicates normal, recurring predecessor which is to be skipped (ignored) when the next run is scheduled into the queues. Only valid when OPT=U and previous value was YES. Automatically reinstated to YES when next successful job completion is done. (See note.) Noted on certain displays as SKIP but does not appear on QM.2 displays since it is skipped when the job enters the queues.

Default: Yes (for OPT value of A)

Required: No

Batch keyword: NEXTRUN

Note: The "next successful job completion" only applies to jobs that are not already in the queue when the NEXT-RUN is set. However, if the job is already in the queue and entered the queue when the NEXT-RUN was SKIP or ONLY, then it is reset as indicated.

4.3.4.2 Usage Notes

You may add additional requirements on a onetime basis. You may skip existing requirements on a onetime basis.

The combination of job name, job schedule ID, network name, and network sub-ID handles any unique variations of a network connection.

A connected input network becomes a requirement for the job. An input network connection is not considered a true requirement for the following conditions:

- The job is scheduled by a RUN or LOAD command.
- The requirement does not apply to this run (schedule ID dependency).
- The requirement was defined as NEXT-RUN=SKIP.
- The job was DEMANDED using TYPE=RES.

A connected output network is scheduled into the postprocess queue when the CPU job enters the request queue. After the network is defined with the DB.5 screen, the network is logically connected to one or more CPU jobs with the DB.3.5 screen. (Required scheduling parameters must also be entered with the DB.2.3 screen; the schedule, however, is not RESOLVED.)

4.3.4.3 Batch Input Example

This batch input example adds output network TESTOTNW as a successor to job CA07XX01. Schedule ID 1 for TESTOTNW is used with station due-out times adjusted by 1 hour. A name of RPT1205 is used to further identify this network from any other which may be associated with the job.

```
JOBCONN  
UPD,NWK,CA07XX01,OPT=A,NETWORK=TESTOTNW,NWKSCHID=1,LEADTM=100,SUBID=RPT1205  
DBM
```


4.3.5.1 Field Descriptions

FUNCTION:	The function to be performed. Value must be the name of some other screen or one of the following:
FORMAT	Clears the screen of user input data.
LIST	Lists data from the database. In batch, a formatted screen is not listed; only a found or not found message is returned.
UPD	Updates a user requirement according to OPT field values.
	Required: Yes
	Batch keyword: Positional parameter
n/a	Predecessor type and has a constant value of USR for user text.
	Required: Yes for batch only
	Batch keyword: Positional parameter
PRED FOR JOB:	The job name for which user requirements are being defined.
	Size/Type: 1 to 8 alphanumeric characters
	Required: Yes
	Batch keyword: Positional parameter
LIST-SCHID:	Applies only to the LIST function. A SCHID value of 0 applies to connections for all schedules and therefore is listed with connections for any schedule ID requested. Schedule IDs on each detail line apply to that line only.
	Size/Type: 1 to 3 numeric characters
	Default: Null (causes connections for all schedule IDs to appear)
	Required: No
	Batch keyword: Not applicable
OPT	This field is used with the UPD function to denote the type of connection operation. The operation codes are A (add), D (delete), and U (update).
	Note: The U option applies only to the NEXT-RUN field. To change the memo text, a delete (D) must be performed followed by an add (A).
	Required: No
	Batch keyword: OPT

SCHID

Indicates the schedule ID (of this job) for which a user requirement is applied. A zero default cannot be specified for one connection and a nonzero schedule ID used for a subsequent connection to the same job with the same user requirement description. An attempt to make such a connection results in an error message issued by CA-7.

Size/Type: 1 to 3 numeric characters from 0 through 255
 Default: 0 (indicates that the requirement holds for all schedule IDs)
 Required: No
 Batch keyword: SCHID

MEMO-FORM USER PREDECESSOR

The free-form description of the user requirement. Not valid with OPT=U. Any text may be entered here that communicates the appropriate requirement. Use this field to define requirements that cannot be controlled automatically, for example ensuring that the transmission line is up or calling the payroll department to verify that the edit is OK. You can express the requirements as a simple phrase, a sentence, or a paragraph. We recommend that you use as many words as you need to clarify exactly what has to be done. For example, "Make sure the online system is down" is not as clear as "Make sure the online system is down for the day" since online systems can come up and down during the day. If the instructions do not fit in the 36 spaces on one line, you can use as many lines as you need to explain them completely. Each line then becomes a separate requirement which must be satisfied before the job can run.

We also recommend that this type of requirement be used only if necessary since manual intervention is required to satisfy it.

Note: If the text contains an apostrophe, parenthesis, or comma, the requirement cannot be added through BTI nor can it be posted through an external communicator.

Size/Type: 1 to 36 alphanumeric characters
 Required: Yes
 Batch keyword: USR

NEXT-RUN

Indicates status of this predecessor for the next run of this job. The next time the job is scheduled into the queues, the predecessors for that execution are determined by the following:

YES Indicates normal, recurring predecessor.

ONLY Indicates onetime predecessor for only the next run of this job. Automatically discarded by CA-7 when next successful job completion is done. (See note.) Only valid when OPT=A. Noted on QM.2 display as O and on various general inquiry displays as ONLY.

SKIP Indicates normal, recurring predecessor which is to be skipped (ignored) when the next run is scheduled into the queues. Only valid when OPT=U and previous value was YES. Automatically reinstated to YES when next successful job completion is done. (See note.) Noted on certain displays as SKIP but does not appear on QM.2 displays since it is skipped when the job enters the queues.

Default: Yes (for OPT value of A)

Required: No

Batch keyword: NEXTRUN

Note: The "next successful job completion" only applies to jobs that are not already in the queue when the NEXT-RUN is set. However, if the job is already in the queue and entered the queue when the NEXT-RUN was SKIP or ONLY, then it is reset as indicated.

4.3.5.2 Usage Notes

You may add additional requirements on a onetime basis. You may skip existing requirements on a onetime basis.

The combination of job name, job schedule ID, and user requirement descriptions handles any unique variations of the user connection.

4.3.5.3 Batch Input Example

This batch input example adds a memo type predecessor requirement for job CA07XX01.

```
JOBCONN
UPD,USR,CA07XX01,OPT=A,USR=CALL FRED BEFORE RELEASING
DBM
```

4.3.6 DB.3.7 - Report IDs Created Screen

Use this screen to record information in the database regarding reports that are produced by CPU jobs. The information is optional and serves only as reference material for anyone wishing to review the information at a terminal. Other means of recording this data, including DB.4 functions, may be more desirable.

[illegible]

To display, enter:

- 7 as the FUNCTION on the DB.3 Menu screen.
- 3.7 as the FUNCTION on the DB Menu screen.
- DB.3.7 as the FUNCTION value on any other menu or formatted input screen.
- DB.3.7 or JOBCONN,RPT as a top line command.

To exit:

- Press the PF3 key to return to the DB.3 Menu screen except with the TSO-ISPF interface when PF3 has been assigned to some other function.
- Move the cursor to the top line of the screen and enter the desired top line command if yet another function is desired.

4.3.6.1 Field Descriptions

FUNCTION:	The function to be performed. Value must be the name of some other screen or one of the following:
FORMAT	Clears the screen of input data.
LIST	Lists data from the database. In batch, a formatted screen is not listed; only a found or not found message is returned.
UPD	Updates report IDs according to OPT field values.
	Required: Yes
	Batch keyword: Positional parameter
n/a	Predecessor type. Has a constant value of RPT for reports.
	Required: Yes for batch only
	Batch keyword: Positional parameter
IDS FROM JOB:	The job name for which report IDs are being defined.
	Size/Type: 1 to 8 alphanumeric characters
	Required: Yes
	Batch keyword: Positional parameter
LIST-SCHID:	Applies only to the LIST function. A SCHID value of 0 applies to connections for all schedules and therefore is listed with connections for any schedule ID requested. Schedule IDs on each detail line apply to that line only.
	Size/Type: 1 to 3 numeric characters
	Default: Null (causes connections for all schedule IDs to appear)
	Required: No
	Batch keyword: Not applicable
OPT	This field is used with the UPD function to denote the type of connection operation. Operation code U (update) cannot be performed on this field. If a change is needed, a delete (D) must be performed followed by an add (A).
	Required: No
	Batch keyword: OPT

SCHID	<p>Indicates the schedule ID of this job for which report IDs are being defined. If omitted when the report ID is defined, default is 0. A zero default cannot be specified for one connection and a nonzero schedule ID used for a subsequent connection to the same job with the same report ID. An attempt to make such a connection results in an error message issued by CA-7. Is not valid with OPT=U.</p> <p>Size/Type: 1 to 3 numeric characters from 0 through 255 Default: 0 (indicates the report IDs apply to all schedule IDs) Required: No Batch keyword: SCHID</p>
REPORT ID	<p>The free-form ID of the report.</p> <p>Size/Type: 1 to 10 alphanumeric characters Required: No Batch keyword: RPTID</p>
STEPNAME	<p>The JCL step name in which the reports are created.</p> <p>Size/Type: 1 to 8 alphanumeric characters Required: No Batch keyword: STEP</p>
DDNAME	<p>The JCL ddname with which the reports are created.</p> <p>Size/Type: 1 to 8 alphanumeric characters Required: No Batch keyword: DD</p>

4.3.6.2 Batch Input Example

This batch input example adds report creation information for job CA07XX01. The report PAYWK4001H is created in step STEP050 from DD statement SYSLIST.

```
JOBCONN
UPD,RPT,CA07XX01,OPT=A,RPTID=PAYWK4001H,STEP=STEP050,DD=SYSLIST
DBM
```


Chapter 5. Virtual Resource Management

Use the Virtual Resource Management (VRM) facility to define controls on job submission based on resource availability. The resource connected to a job can represent a real resource such as a data set or a started task.

This facility provides the following features:

- Resource control at the job, system, or step level.
- Job submission control for jobs that use shared or exclusive resources.
- Job corequisite requirements.

Note: The Virtual Resource Management (VRM) facility manages resource utilization through the use of tables internal to CA-7.

Resource dependencies are determined during the job submission process. Jobs waiting for resources will wait in the CA-7 ready queue with a status of W-RSRC.

VRM resources are ignored for non-executable jobs.

5.1 Resource Types

5.1.1 Shared Resources

There may be times when multiple jobs run using the same resource. You can define this as a Shared resource requirement under the Virtual Resource Management facility. A Shared resource connection indicates that multiple jobs can execute at the same time while using this resource.

5.1.2 Exclusive Resource

You can define the exclusive use of a resource under CA-7 VRM. This indicates that the job requires exclusive use of the resource and that no other jobs connected to the resource may execute. If the 'resource' is unavailable exclusively at job submission time, the job waits in the CA-7 ready queue for the resource with a status of W-RSRC.

5.1.3 Corequisite Resources

A corequisite resource relationship indicates that a job executes based on the active or inactive status of the resource. If a job to resource connection is set up to require a specific corequisite resource to be active prior to execution, CA-7 checks whether the corequisite resource is active prior to submitting the job. The same is true for an inactive corequisite resource. The indicated corequisite resource must be inactive before the job is submitted for execution. To activate or deactivate a corequisite resource, issue the PRSQA and PRSQD commands respectively. You can do this in an online or batch environment.

Note: The following corequisite resource name prefixes are reserved:

FLOW@
CPM@

5.1.4 Address Space Resources

The CA-7 Virtual Resource Management facility provides the ability to control the execution of a job based on the active or inactive status of an address space. For example, assume that JOBA requires that a CICS region be active for the job to be submitted. During the job submission cycle, CA-7 checks to see if the CICS region is active prior to submitting the job. If the region is not active, the job waits in the CA-7 ready queue with a status of W-RSRC until the address space becomes active. This check occurs during every CA-7 submission cycle. The same process occurs for a job connected to an address space resource that requires that the address space be inactive. The address space resource requirement is only valid on the host CPU where CA-7 is executing.

5.1.5 Resource Count Resources

A resource count resource is a resource which is available for use based on a specific count or number for this resource. The count value indicates the maximum occurrences for this resource count resource. When a job to resource connection is made to a resource count resource, a value is specified which identifies the total number of occurrences for this resource that the job will use. When the job is submitted, this value is subtracted from the maximum amount available for the resource count resource.

For example, suppose that you have 20 tapes drives available at your site and you wish to manage the usage of the tape drives by connecting jobs to a resource count resource. The resource count resource can indirectly represent the tape drives by establishing a maximum total count available for the resource count resource and then connecting jobs to the resource and specifying the number of occurrences that each job will use.

Note: CA-7 reserves the following resource count resource name prefix:

VRMDD.

5.2 Handling Resource Conflicts

VRM dynamically manages resource conflicts based on VRM database definitions and CA-7 job queue activity. This usually requires little or no manual intervention. CA-7 detects when a job is no longer using a resource and frees it for use by other jobs according to the instructions in the jobs resource profile (the FREE value on the RM.1 panel). However anomalies may occur if there are multiple instances of a job with the same name.

For example, suppose JOBA is defined to use resource RSC1. Further assume that the RM.1 definition specifies TYPE=EXC and FREE=Y. If JOBA (say, job number 1) abends and is waiting in the Request Queue for restart, another instance of JOBA (job number 2) may move to the ready queue for submission. However, JOBA (1) still has exclusive control of the RSC1 resource that was acquired when it was initially submitted. Hence, JOBA (2) will not submit. Instead it will wait in the ready queue for the RSC1 resource to become available. An LQ display will show that the status of JOBA (2) is W-RSRC.

In order to allow JOBA (2) to run, one must take manual corrective action such as canceling or forcing completion of JOBA (1). JOBA (1) cannot be submitted while JOBA (2) is in the ready queue.

An initialization file option is provided to streamline the handling of such conflicts.

See the discussion of the OPTIONS statement keyword JSOP in the *CA-7 Systems Programmer Guide* for information on handling these anomalies.

5.3 VRM Menu Screen

Use the RM (Resource Management) Menu screen to select the Virtual Resource Management functions.

```

----- CA-7 VIRTUAL RESOURCE MANAGEMENT MENU -----
FUNCTION ==>

RESOURCE MANAGEMENT OPTIONS:

  1 - JOB RESOURCE MANAGEMENT
  2 - JOB RESOURCE CROSS REFERENCE LIST
  3 - ACTIVE JOB RESOURCES DISPLAY
  4 - PENDING RESOURCES JOB DISPLAY
  5 - JOBS WAITING ON RESOURCES DISPLAY
  6 - COREQUISITE RESOURCES LIST
  7 - RESOURCE COUNT RESOURCE MANAGEMENT

PROGRAM: RSC0 MSG-INDX: 00 -- RM -- yy.266 / 13:21:42
MESSAGE: SPECIFY DESIRED OPTION OR ENTER A COMMAND ON THE TOP LINE

```

To display, enter:

- RM as a top line command.
- RM as the FUNCTION value on any other menu or formatted input screen.

To exit, enter:

- The name of an online screen as the FUNCTION value or move the cursor to the top line and enter a top line command if some other function is desired.

5.3.1 Usage Notes

Select the desired function by entering its value as the FUNCTION and pressing the Enter key.

5.4 RM.1 Job Resource Management Screen

Use this screen to list, add, update, or delete resource connections by job. It generates the VRM static type J and R records maintained on the VRM database component.

```

----- CA-7 JOB RESOURCE MANAGEMENT -----
FUNCTION =====>          (FORMAT,LIST,UPD)
JOB:                               LIST-SCHID:          PAGE 0001
OPT SCHID   RESOURCE NAME          STEPNAME   TYPE FREE

OPTIONS: A=ADD  D=DELETE  U=UPDATE  *=PROCESSED  ?=ERROR
PROGRAM: RSC1  MSG-INDX: 00  -- RM.1    -- yy.ddd / hh:mm:ss
MESSAGE: ENTER FUNCTION, TRANSFER OR ENTER A COMMAND ON THE TOP LINE

```

To display, enter:

- 1 as the FUNCTION on the RM Menu screen.
- RM.1 as the FUNCTION value on any other menu or formatted input screen.
- RM.1 as a top line command.

To exit:

- Press the PF3 key to return to the RM Menu screen except with the TSO/ISPF interface when PF3 has been assigned to some other function.
- Move the cursor to the top line of the screen and enter the desired top line command if yet another function is desired.

5.4.1 Field Descriptions

FUNCTION The function to be performed. Value must be the name of some other screen or one of the following:

FORMAT Clears the screen of user input data.

LIST Lists the resources connected to the specified job and schedule ID.

UPD Performs updating option indicated by the OPT parameter below.

Required: Yes

Batch keyword: Positional parameter

JOB: The job name on which the indicated function is to be performed.

Size/Type: 1 to 8 alphanumeric characters

Required: Yes

Batch keyword: JOB

LIST-SCHID:

Applies only to the LIST function. A SCHID value of 0 applies to connections for all schedules and therefore is listed with connections for any schedule ID requested. Schedule IDs on each detail line apply to that line only.

Size/Type: 1 to 3 numeric characters from 0 to 255

Default: Null (causes connections for all schedule IDs to appear)

Required: No

Batch keyword: Not applicable

OPT This field is used with the UPD function to denote the type of connection operation. The operation codes are A (add), U (update), and D (delete).

Required: Yes

Batch keyword: OPT

SCHID Indicates the schedule ID (of this job) for which a user requirement is applied. A zero default cannot be specified for one connection and a nonzero schedule ID used for a subsequent connection to the same job with the same user requirement description. An attempt to make such a connection results in an error message issued by CA-7.

Size/Type: 1 to 3 numeric characters from 0 through 255

Default: 0 (indicates that the requirement holds for all schedule IDs)

Required: No

Batch keyword: SCHID

RESOURCE NAME

The resource name being connected to the job.

Size/Type: 1 to 44 characters - the first character must be nonblank and no embedded blanks

Required: Yes

Exception: If the resource connection is made to a resource count resource, then the resource name can be from 1 to 39 characters followed by a / (slash) and a 1 to 4 numeric value indicating the number of occurrences for the resource that the job will use.

Batch keyword: RSRC

STEPNAME

The resource will be freed at the conclusion of the named job step. Not valid for types ASX and CRQ.

Size/Type: 1 to 8 alphanumeric characters

Required: No

Batch keyword: STEP

Note: The stepname field applies to the first occurrence of the stepname encountered in the job stream. This includes any reference to the stepname embedded in a PROC.

TYPE

This field determines the usage of the resource by the job. The valid resource usage types are:

ASX An address space resource. This controls the submission of the job based on the status of an address space.

Note: The address space resource requirement is only valid on the host CPU where CA-7 is executing.

CRQ A corequisite resource. The job cannot be submitted if the resource is active or inactive (see the FREE option). The corequisite resource must be activated or deactivated using the PRSQA or PRSQD command. This can be performed in an online or batch environment.

EXC An exclusive resource. Only one job can be active with this resource.

RCT A resource count resource. This controls the submission of the job based on the number of available occurrences within a resource count type resource when adding a resource count resource connection. The resource name can be from 1 to 39 characters followed by a / (slash) and a 4-character numeric value indicating the number of resource occurrences this job will use.

SHR A shared resource. Multiple jobs can be active with the same resource.

Required: Yes

Batch keyword: TYPE

FREE

The FREE option determines how VRM manages resource availability at job submission and job/step completion. Different values are supported depending upon the type of resource defined.

For shared and exclusive resources the following FREE options are supported. There is no default.

- A** The resource will only be freed if any step in the job abends (abend or condition code checking.)
- F** The resource will be freed when job ends, either successfully or unsuccessfully. If the STEPNAME parameter is specified, free the resource when the specified step completes, either successfully or unsuccessfully.
- N** Do not free the resource at successful job completion. The resource may be freed with the PRSCF command. A cancel of the job also frees the resource.
- Y** Free the resource at successful job completion. If the STEPNAME parameter is specified, free the resource when the specified step completes successfully (does not abend).

For corequisite resources the following FREE options are supported:

- A** The job will be submitted only if the resource is active (as established by the PRSQA command.) This is the default.
- I** The job will be submitted only if the resource is NOT active.

For address space resources the following FREE options are supported:

- A** The job will be submitted only if the resource is active on the same system as CA-7. The resource name is assumed to be a jobname or started task name running on the system. This is the default.
- I** The job will be submitted only if the resource is NOT active on the same system as CA-7. The resource name is assumed to be a jobname or started task name running on the system.

For resource count resources the following FREE options are supported. There is no default.

- A** The resource count will only be decremented when the job abends.
- F** The resource count will be decremented when the job completes, either successfully or unsuccessfully. If the STEPNAME parameter is specified, the resource count is decremented when the specified step completes, either successfully or unsuccessfully.

N	The resource count is not decremented when the job completes. The PRSCF command may be used to decrement the resource count. A cancel of the job also decrements the resource count.
Y	The resource count will be decremented when the job or step ends successfully. If the STEPNAME parameter is specified, the resource count is decremented when the specified step completes successfully (does not abend).
Required:	Yes
Batch keyword:	FREE

5.4.2 Batch Input Example

This batch input example lists the resources connected to job TESTJOB.

```
RM.1
LIST, RM.1, JOB=TESTJOB
```

This batch input example adds a resource connection to job TESTJOB.

```
RM.1
UPD, RM.1, JOB=TESTJOB, OPT=A, RSRC=SAMPLE.RESOURCE, SCHID=0, TYPE=SHR, FREE=F
```

5.4.3 Critical Path Monitoring Corequisite Resources

You can use CA-OPS/MVS II with CA-7 to monitor the progress and status of critical paths in your workload.

To define a critical path flow, you need to connect a unique type of VRM corequisite resource to the first job in the flow. This resource defines the name, end point, and target time for the flow. When the job is run, an active flow element is created when VRM resources are attached in the ready queue.

The format of the resource name is:

```
FLOW@flowname.endjob.endschid.endtarget.endday
```

Where:

flowname

The 1-8 character unique name to assign to this flow.

endjob

The job name of the last job in this flow.

endschid

The schedule ID of the last job in this flow.

endtarget

The target time that the ending job should complete by (hhmm).

endday

An optional parameter to add day(s) to the target time (.n).

5.4.3.1 Example

FLOW@PAYFLOW1.PAYJOBZ.1.1700

5.4.3.2 Notes

1. The resource type must be corequisite (CRQ).
2. The resource name must begin with FLOW@.
3. The flowname, end job name, end schedule ID, and end target are required parameters and must be separated by periods.
4. The end day parameter is optional. If specified it must be 1 digit (0-9) separated from the end target by a period. A value of 1 will cause 24 hours to be added to the target time. A value of 2 will cause 48 hours to be added, and so forth.
5. The 'FREE' type must be specified on the definition (A or I), however, it is ignored if the CPM facility is active. Since these resources are treated as normal corequisites if the CPM facility is not active, you may wish to specify I (inactive) for the FREE type.
6. VRM RESOURCES ARE NEVER ATTACHED TO NON-EXECUTABLE JOBS.
Thus, the starting job of any flow must be executable.

See the *CA-7 Interfaces Guide* and the CA-OPS/MVS II documentation for more information on the Critical Path Monitoring facility.

5.5 RM.2 Job/Resource Cross Reference List Screen

Use this screen to list jobs using resources. The jobs can be listed by a fully qualified resource name or by a generic key.

```

----- CA-7 JOB / RESOURCE CROSS REFERENCE LIST -----
FUNCTION =====>          (FORMAT,LIST)
RSRC:                      LIST-SCHID:      PAGE 0001
  RESOURCE NAME                SCHID JOBNAME  STEPNAME TYPE FREE

```

```

PROGRAM: RSC2  MSG-INDX: 00  -- RM.2      --  yy.ddd / hh:mm:ss
MESSAGE: ENTER FUNCTION, TRANSFER OR ENTER A COMMAND ON THE TOP LINE

```

To display, enter:

- 2 as the FUNCTION on the RM Menu screen.
- RM.2 as the FUNCTION value on any other menu or formatted input screen.
- RM.2 as a top line command.

To exit:

- Press the PF3 key to return to the RM Menu screen except with the TSO/ISPF interface when PF3 has been assigned to some other function.
- Move the cursor to the top line of the screen and enter the desired top line command if yet another function is desired.

5.5.1 Field Descriptions

FUNCTION	The function to be performed. Value must be the name of some other screen or one of the following: FORMAT Clears the screen of user input data. LIST Lists resources and the associated jobs based on user specified resource criteria. Required: Yes Batch keyword: Positional parameter
RSRC:	Specifies the resource(s) for which information is to be listed. * Indicates all resources. resource Indicates a fully qualified resource name. Size/Type: 1 to 44 alphanumeric characters resource* Indicates multiple resources specified by a generic resource name. Size/Type: 1 to 43 alphanumeric characters terminated with an asterisk Required: Yes Batch keyword: RSRC
LIST-SCHID:	Applies only to the LIST function. A SCHID value of 0 applies to connections for all schedules and therefore is listed with connections for any schedule ID requested. Schedule IDs on each detail line apply to that line only. Size/Type: 1 to 3 numeric characters from 0 to 255 Default: Null (causes connections for all schedule IDs to appear) Required: No Batch keyword: SCHID
RESOURCE NAME	Displays the requested resources.
SCHID	Displays the schedule ID of the job using this resource.
JOBNAME	The name of the job using this resource.
STEPNAME	The name of the step that frees this resource.
TYPE	The usage mode of the resource (for example, SHR, EXC).
FREE	The disposition of this resource at step, abnormal, or successful completion time. See the FREE field on page 5-9 for more information.

5.5.2 Batch Input Example

This batch input example lists all the cross-reference entries for resources that begin with PROD.

```
RM.2  
LIST, RM.2, RSRC=PROD*
```

5.6 RM.3 Active Job Resources Display Screen

Use this screen to display active resources by job. The jobs can be listed generically or with fully qualified job names.

```

----- CA-7 ACTIVE JOB RESOURCES DISPLAY -----
FUNCTION ==>> (FORMAT,LIST)
JOB:          RSRC:          PAGE 0001
JOBNAME RESOURCE NAME      CA7 # STEPNAME TYPE FREE

PROGRAM: RSC3  MSG-INDX: 00  -- RM.3    -- yy.ddd / hh:mm:ss
MESSAGE: ENTER FUNCTION, TRANSFER OR ENTER A COMMAND ON THE TOP LINE

```

To display, enter:

- 3 as the FUNCTION on the RM Menu screen.
- RM.3 as the FUNCTION value on any other menu or formatted input screen.
- RM.3 as a top line command.

To exit:

- Press the PF3 key to return to the RM Menu screen except with the TSO/ISPF interface when PF3 has been assigned to some other function.
- Move the cursor to the top line of the screen and enter the desired top line command if yet another function is desired.

5.6.1 Field Descriptions

FUNCTION	<p>The function to be performed. Value must be the name of some other screen or one of the following:</p> <p>FORMAT Clears the screen of user input data.</p> <p>LIST Displays active resources associated with user specified jobs.</p> <p>Required: Yes Batch keyword: Positional parameter</p>
JOB	<p>Specifies a fully qualified or generic job name.</p> <p>* Indicates all jobs.</p> <p>job Indicates a fully qualified job name.</p> <p>Size/Type: 1 to 8 alphanumeric characters</p> <p>job* Indicates multiple jobs specified by a generic job name.</p> <p>Size/Type: 1 to 7 alphanumeric characters terminated with an asterisk</p> <p>Default: All jobs Required: No Batch keyword: JOB</p>
RSRC	<p>Specifies a fully qualified or generic resource name. An asterisk is used to indicate a generic request.</p> <p>* Indicates all resources</p> <p>resource Indicates a fully qualified resource name.</p> <p>Size/Type: 1 to 44 alphanumeric characters</p> <p>resource* Indicates multiple resources specified by a generic resource name.</p> <p>Size/Type: 1 to 43 alphanumeric characters terminated with an asterisk</p> <p>Default: All resources Required: No Batch keyword: RSRC</p>
JOBNAME	Displays the job name.
RESOURCE NAME	Displays the names of the active resources.
CA7 #	The CA-7 job number.
STEPNAME	The name of the job step that freed the resource.
TYPE	The usage mode of the resource for the job.

FREE

The disposition of the resource at job completion time or the status of a corequisite resource type.

5.6.2 Batch Input Example

This batch input example lists all active resources for jobs that begin with PAY.

```
RM.3  
LIST, RM.3, JOB=PAY*, RSRC=*
```

5.7 RM.4 Pending Resources Job Display Screen

Use this screen to display all pending resources with an associated job name. A pending resource is a nonfreed resource connected to a job which has already executed and has been purged from the request queue. This resource can only be freed by using the PRSCF command. The pending resources can be listed using a generic or fully qualified resource name.

```

----- CA-7 PENDING RESOURCES JOB DISPLAY -----
FUNCTION =====>          (FORMAT,LIST)
RSRC:
RESOURCE NAME                                JOBNAME CA7 JOB# TYPE
                                           PAGE 0001

PROGRAM: RSC4  MSG-INDX: 00  -- RM.4      -- yy.ddd / hh:mm:ss
MESSAGE: ENTER FUNCTION, TRANSFER OR ENTER A COMMAND ON THE TOP LINE

```

To display, enter:

- 4 as the FUNCTION on the RM Menu screen.
- RM.4 as the FUNCTION value on any other menu or formatted input screen.
- RM.4 as a top line command.

To exit:

- Press the PF3 key to return to the RM Menu screen except with the TSO/ISPF interface when PF3 has been assigned to some other function.
- Move the cursor to the top line of the screen and enter the desired top line command if yet another function is desired.

5.7.1 Field Descriptions

FUNCTION	<p>The function to be performed. Value must be the name of some other screen or one of the following:</p> <p>FORMAT Clears the screen of user input data.</p> <p>LIST Displays all pending resources (nonfreed) and the associated job name. A pending resource may be freed using the PRSCF command.</p> <p>Required: Yes</p> <p>Batch keyword: Positional parameter</p>
RSRC	<p>Specifies a fully qualified or generic resource name. An asterisk is used to indicated a generic request.</p> <p>* Indicates all resources.</p> <p>resource Indicates a fully qualified resource name.</p> <p>Size/Type: 1 to 44 alphanumeric characters</p> <p>resource* Indicates multiple resources specified by a generic resource name.</p> <p>Size/Type: 1 to 43 alphanumeric characters terminated with an asterisk</p> <p>Default: All resources</p> <p>Required: No</p> <p>Batch keyword: RSRC</p>
RESOURCE NAME	Displays the pending resource names.
JOBNAME	Displays the job name.
CA7 JOB#	Displays the CA-7 job number.
TYPE	Displays the usage mode of the resource by this job.

5.7.2 Batch Input Example

This batch input example lists all pending VRM resources.

```
RM.4
LIST, RM.4, RSRC=*
```

5.8 RM.5 Jobs Waiting on Resources Screen

Use this screen to display jobs waiting for resources. It can be used to show the resources that a job is waiting for when the job shows a status of W-RSRC on the List Queue (LQ) display. However, ASX resources may not always show on the RM.5 screen. This can occur because ASX resources are only checked periodically (approximately two minute intervals) after the initial test for a job.

```

----- CA-7 JOBS WAITING ON RESOURCES -----
FUNCTION =====>      (FORMAT,LIST)
JOB:                      LIST-SCHID:                      PAGE 0001
JOB NAME  RESOURCE NAME                                SCHID STEPNAME TYPE FREE

```

```

PROGRAM: RSC5  MSG-INDX: 00  -- RM.5      -- yy.ddd / hh:mm:ss
MESSAGE: ENTER FUNCTION, TRANSFER OR ENTER A COMMAND ON THE TOP LINE

```

To display, enter:

- 5 as the FUNCTION on the RM Menu screen.
- RM.5 as the FUNCTION value on any other menu or formatted input screen.
- RM.5 as a top line command.

To exit:

- Press the PF3 key to return to the RM Menu screen except with the TSO/ISPF interface when PF3 has been assigned to some other function.
- Move the cursor to the top line of the screen and enter the desired top line command if yet another function is desired.

5.8.1 Field Descriptions

FUNCTION	The function to be performed. Value must be the name of some other screen or one of the following: FORMAT Clears the screen of user input data. LIST Displays jobs waiting for resources based on user specified job name and schedule ID criteria. Required: Yes Batch keyword: Positional parameter
JOB	Specifies a fully qualified or generic job name. * Indicates all jobs. job Indicates a fully qualified job name. Size/Type: 1 to 8 alphanumeric characters job* Indicates multiple jobs specified by a generic job name. Size/Type: 1 to 7 alphanumeric characters terminated with an asterisk Default: All jobs Required: No Batch keyword: JOB
LIST-SCHID	Applies only to the LIST function. A SCHID value of 0 applies to connections for all schedules and therefore is listed with connections for any schedule ID requested. Schedule IDs on each detail line apply to that line only. Size/Type: 1 to 3 numeric characters from 0 to 255 Default: Null (causes connections for all schedule IDs to appear) Required: No Batch keyword: SCHID
JOB NAME	Displays the job name of the job awaiting the resource.
RESOURCE NAME	Displays the name of the nonavailable resource.

SCHID	Displays the schedule ID of the job awaiting the resource.
STEPNAME	Displays the jobstep in which the resource is to be freed.
TYPE	Displays the usage mode of the resource by the job. Note: On this screen, a TYPE of FRE may show. This would indicate that a PRSCF command with FORCE=YES has been used for this resource.
FREE	Displays the disposition of the resource at job completion time.

5.8.2 Batch Input Example

This batch input example lists jobs waiting on VRM resources.

```
RM.5  
LIST, RM.5, JOB=*
```

5.9 RM.6 Corequisite Resources List Screen

Use this screen to list active corequisite resources.

```

----- CA-7 COREQUISITE RESOURCES LIST -----
FUNCTION ==> (FORMAT,LIST)
RSRC:
CO-REQ RESOURCE NAME
PAGE 0001

PROGRAM: RSC6 MSG-INDX: 00 -- RM.6 -- yy.ddd / hh:mm:ss
MESSAGE: ENTER FUNCTION, TRANSFER OR ENTER A COMMAND ON THE TOP LINE

```

To display, enter:

- 6 as the FUNCTION on the RM Menu screen.
- RM.6 as the FUNCTION value on any other menu or formatted input screen.
- RM.6 as a top line command.

To exit:

- Press the PF3 key to return to the RM Menu screen except with the TSO/ISPF interface when PF3 has been assigned to some other function.
- Move the cursor to the top line of the screen and enter the desired top line command if yet another function is desired.

5.9.1 Field Descriptions

FUNCTION

The function to be performed. Value must be the name of some other screen or one of the following:

FORMAT Clears the screen of user input data.

LIST Displays active corequisite sources based on user specified job name and schedule ID criteria.

Required: Yes

Batch keyword: Positional parameter

RSRC: Resource name

* Indicates all resources.

resource Indicates a fully qualified resource name.

Size/Type: 1 to 44 alphanumeric characters

resource* Indicates multiple resources specified by a generic resource name.

Size/Type: 1 to 43 alphanumeric characters terminated with an asterisk

Required: Yes

Batch keyword: RSRC

CO-REQ RESOURCE NAME

The fully qualified corequisite resource name.

5.9.2 Batch Input Example

This batch input example lists all active corequisite resources.

```
RM.6  
LIST, RM.6, RSRC=*
```

5.10 RM.7 Resource Count Resource Management Screen

Use this screen to list, add, update, or delete a resource count type resource.

```

----- CA-7 RESOURCE COUNT RESOURCE MANAGEMENT -----
FUNCTION =====>          (FORMAT,LIST,UPD)
RSRC:
OPT  RESOURCE NAME                                TOTAL COUNT    PAGE NO. 0001
                                           CURR IN USE

OPTIONS: U=UPDATE A=ADD D=DELETE   *=PROCESSED  ?=ERROR
PROGRAM: RSC7  MSG-INDX: 00  -- RM.7      -- yy.115 / 08:16:29
MESSAGE: ENTER FUNCTION, TRANSFER OR ENTER A COMMAND ON THE TOP LINE

```

To display, enter:

- 7 as the FUNCTION on the RM Menu screen.
- RM.7 as the FUNCTION value on any other menu or formatted input screen.
- RM.7 as a top line command.

To exit:

- Press the PF3 key to return to the RM Menu screen except with the TSO/ISPF interface when PF3 has been assigned to some other function.
- Move the cursor to the top line of the screen and enter the desired top line command if yet another function is desired.

5.10.1 Field Descriptions

FUNCTION	<p>The function to be performed. Value must be the name of some other screen or one of the following:</p> <p>FORMAT Clears the screen of all input data.</p> <p>LIST Lists the Resource Count Resource(s) and related resource information.</p> <p>UPD Used to update information about the Resource Count Resource.</p> <p>Required: Yes Batch keyword: Positional parameter</p>
RSRC:	<p>Indicates a fully qualified or generic Resource Count Resource name. This field is only valid with LIST function.</p> <p>* Indicates all resources.</p> <p>resource Indicates a fully qualified resource name. Size/Type: 1 to 39 alphanumeric characters</p> <p>resource* Indicates multiple resources specified by a generic resource name. Size/Type: 1 to 38 alphanumeric characters terminated with an asterisk</p> <p>Required: Yes Batch keyword: RSRC</p>
OPT	<p>Used with the Update and Add functions to denote the operation to perform. Valid codes are U (update), A (add), and D (delete).</p> <p>Batch keyword: OPT</p>
RESOURCE NAME	<p>Indicates the Resource Count Resource for which the option is to be performed.</p> <p>Batch keyword: DSN</p>
TOTAL COUNT	<p>Indicates the total number of available occurrences for this Resource Count Resource.</p> <p>Batch keyword: TOTAL</p>
CURR IN USE	<p>Indicates the number of occurrences of this Resource Count Resources currently in use.</p>

5.10.2 Batch Input Example

This batch input example adds a resource name of PAYROLL.RCT with a total of 100.

```
RM.7  
UPD, RM.7, OPT=A, RSRC=PAYROLL.RCT, TOTAL=100
```

This batch input example lists all resource count resources that are currently defined.

```
RM.7  
LIST, RM.7, RSRC=*
```

5.11 VRM Device Control

5.11.1 Overview

VRM Device Control provides a way to control job submission based on the availability of devices that are detected during the CA-7 Database Load process. With this option, CA-7 defines VRM Resource Count Resources that correspond to real devices used by the job. RM.7 can then be used to set the quantities of available units.

A VRM device is defined using a device code and unit name combination. The Database Load process extracts the device code and/or unit name from each DD statement in the job. Entries in the SASSDTAB table are used to determine which device code/unit name combinations are eligible to be treated as VRM devices. The load process then defines a VRM resource count resource for each unique device code/unit name combination that is allowed by SASSDTAB.

Each entry in SASSDTAB contains information on a device code/unit name combination and the name of a group to which the device belongs. This name is chosen at the discretion of the client and allows definition of resource pools. When the job is submitted, the name of each VRM device is changed to reflect the resource groupings defined in SASSDTAB. Thus, although VRM devices are defined at the device level, they are monitored and controlled at the group level based upon the definitions in SASSDTAB.

5.11.2 VRM Device Control Functions - Definition Structure

CA-7 treats a resource count resource definition as a VRM device definition based upon the structure of the name. The name of a resource count resource that is used as a VRM device definition conforms to the following format:

```
VRMDD.Dxxxxxxx.Uyyyyyyy/nnnn
```

where xxxxxxxx is the device code in character hex, yyyyyyyy is the UNIT name and nnnn is the number of references to this device code and UNIT combination calculated during the CA-7 Database LOAD process. If the device code is not available, a value of 00000000 is used. If the unit name is not available, no value is used.

Each of the following is a valid VRM device definition:

```
VRMDD.D12345678.UROUND/0030
VRMDD.D00000000.UCART/0010
VRMDD.D00000000.U/0050
```

VRM device definitions may be listed on the job's RM.1 panel. The term 'VRMDD.' begins each definition. Since VRM resource names may have already been defined that begin with 'VRMDD.' it is recommended that RM.2 be used to list all VRM resources that begin with 'VRMDD.*' to ensure that resource names will not conflict prior to implementation of the feature.

5.11.3 VRM Device Control Functions - Definition Control

Definitions for VRM Devices may be entered manually using the RM.1 panel. However, it is recommended that VRM devices be defined dynamically. Dynamic definition is indicated by VRMDD=D or VRMDD=Y on the OPTIONS statement in the CA-7 initialization file. With dynamic VRM device definition, CA-7 adds or deletes the resource count resource definitions that correspond to devices detected by SASSJJCL during the LOAD completion process.

The CA-7 Database Load program (SASSJJCL) scans each DD statement in the job and extracts device code and unit name information. The SASSDTAB table is used to determine which device code/unit name combinations are eligible to be considered VRM Devices. If no entry is found in SASSDTAB for the device code/unit name, no resource count resource is created for the device.

Entries in the SASSDTAB table are created using the L2VDD macro. Each entry declares a device code and unit name combination that is to be tracked using VRM device definitions.

SASSJJCL selects devices for dynamic definition based upon a scan of the SASSDTAB table from top to bottom. Thus, selection depends on the order of L2VDD macro statements. L2VDD requires a group name. The DEVICE and UNIT parameters are optional. The value of the DEVICE and UNIT parameters is considered fully qualified unless an asterisk (*) is coded. If the asterisk is coded, any value is considered eligible for dynamic definition.

The following example illustrates significant features of the SASSDTAB module:

	column 10	16	72
SASSDTAB	CSECT		
	SASSVRSN	VRSN=DOC	
ENTRY1	L2VDD	GROUP=SYSDISK, DEVICE=3033000E, UNIT=SYSDA	+ +
*			
ENTRY2	L2VDD	GROUP=ANYDISK, DEVICE=3033000F, UNIT=*	+ +
*			
ENTRY3	L2VDD	GROUP=SOME3380, DEVICE=*, UNIT=3380	+ +
*			
ENTRY4	L2VDD	GROUP=ANYUNIT, DEVICE=*, UNIT=*	+ +
*			
ENTRY5	L2VDD	GROUP=LAST	
	END		

SAMPJCL includes a SASSDTAB sample. It may be located by finding the entry for VRM Device Control in the \$\$INDEX member.

Note that L2VDD macro statements that include an asterisk are placed lower in the module. In this example, all devices detected with a device code of 3033000E and a UNIT name of SYSDA are selected for definition and associated with the group SYSDISK. ENTRY2 ensures that devices with a device code 3033000F are selected for definition regardless of UNIT name value. Such devices will be placed in the ANYDISK group. A device that is not considered eligible according to the aforementioned criteria will be selected for definition if the UNIT name of 3380 is detected for it by SASSJCL. It will be placed in the SOME3380 group. ENTRY4 in the table ensures that any device not already covered by a L2VDD entry will be selected for definition and placed in the ANYUNIT group.

L2VDD GROUP=LAST terminates the table and is required. 'LAST' is a reserved group name.

5.11.4 VRM Device Control Functions - Submission Control

If VRMDD=Y or VRMDD=S is coded on the OPTIONS statement in the CA-7 initialization file, jobs are not submitted unless all required VRM devices are available. Each VRM device definition is modified based on the GROUP name that is associated with the device in SASSTAB. For example, after a job is LOAded, its RM.1 panel displays the following entries:

```
VRMDD.D3390000F.UDISK/0001
VRMDD.D3380000E.USYSDA/0001
```

When VRM attempts to submit the job, it associates all of the resources according to the groups defined in SASSTAB. Thus these entries are reduced to one:

```
VRMDD.ANYUNIT/0002
```

This is the resource name that is displayed on RM.3 and RM.5. RM.7 must be updated with sufficient quantities of the VRMDD.ANYUNIT resource for the job to be submitted.

5.11.5 VRM Device Control - Activation

VRM device control comprises two functions: definition and submission control. Either or both of these functions may be activated depending on installation needs. Activation of these functions is indicated by keyword values on the OPTIONS statement in the CA-7 initialization file.

If VRMDD=D is specified, VRM devices are dynamically defined during LOAD completion, however they are ignored during job submission. If VRMDD=S is specified, no VRM devices are dynamically defined, however any existing VRM device definitions are respected during job submission. VRMDD=Y indicates that both functions are in effect.

By default, VRM devices are not dynamically defined nor are they modified during submission. Unless VRM device control is indicated by VRMDD=Y or S, any RCT resource name beginning with 'VRMDD.' is ignored during submission.

Chapter 6. Automated Recovery Facility

The Automated Recovery Facility (ARF) for CA-7 can be used to monitor exception conditions for production jobs and to schedule recovery actions to execute at or near the point of failure.

Kinds of exception conditions monitored by ARF include but are not limited to:

- Abend exceptions tested at the job or step level.
- Condition code exceptions tested at the job or step level.
- Jobs whose elapsed execution time falls outside a range specified by the user.
- Jobs considered late according to CA-7.
- Jobs considered late according to user specified criteria tested when the job begins or completes.

The kinds of recovery actions that may be executed in response to the exception conditions detected by ARF include:

- Scheduling and tracking special recovery jobs.
- Issuing special messages to a specified TSO user or to the MVS console.
- Issuing CA-7 or MVS commands.
- Restarting, canceling, or forcing the completion of jobs as the final step in the recovery for job completion exceptions.

6.1 Monitoring Exception Conditions

Because recovery monitoring may be resource intensive, ARF was designed so that the recovery needs of a production job are tested at a limited number of points during CA-7 processing. These tests are executed only if they are specifically indicated in the definitions of ARF conditions that are associated with the job when it enters the request queue. These points include: job submission, SMF feedback, late prompting, and job completion.

The definitions that provide the criteria for ARF tests are bundled into groups known as ARFSETs. All of the tests that ARF will use to monitor the status of a job are determined by the ARFSET that is associated with the job when the job enters the request queue.

An ARFSET may be associated with a job in several ways. An ARFSET can be named on the DB.1 panel. This can be overridden using the #ARFSET statement in the JCL. Both of these designations can be overridden using the ARFSET keyword on the DEMAND command.

6.2 Responding to Exception Conditions

Recovery actions are scheduled when the exception is detected so that the action may be performed at or near the point of failure. Each occurrence of an exception condition may have up to 7 distinct actions associated with it in the definition of the condition. The order in which the actions occur in the definition of the condition determines the order in which actions are executed.

Each exception is handled in the order that it is detected. Because recovery actions are executed serially for a given job, all of the actions for an exception occurrence must be completed before the actions of a later exception occurrence can be executed. Suppose, for example, that one step completion condition is detected and has two CA-7 commands associated with it. ARF will begin processing the commands as soon as the exception is recognized. Also suppose that another step completion exception is detected for the same job while ARF is issuing commands for the first exception. In this case, responses associated with the second exception will not be executed until those associated with the first exception have been handled.

Responses for an ARF exception are executed on an internal terminal dedicated for use by ARF. Each response is translated into a CA-7 terminal command or set of such commands. Since all CA-7 terminal command input is logged to the CA-7 Browse Data Set, all ARF recovery activity is thereby recorded.

6.3 Types of Exceptions Recognized by ARF

All ARF monitoring is organized by exception type. Each ARF definition in an ARFSET must specify the type of exception that is to be monitored. A list of the types of exceptions recognized by ARF follows.

EC - Elapsed Time Check at Completion

A check of the elapsed time for the job is made when CA-7 is notified of normal job completion. The ARF definition specifies an elapsed time value (in minutes) and a relational operator. If the elapsed time of the completed job stands in the relation indicated by the relational operator to the elapsed time value in the ARF definition, then an ARF exception is recognized for the job. Thus, for example, if the ARF definition associated with a job specifies a relational operator of 'GE' and an elapsed time value of '0002', then an ARF exception will be recognized if the elapsed time at completion of the job is greater than or equal to 2 minutes.

EE - Elapsed Time Check During Execution

The elapsed time of the job is monitored during execution. The ARF definition specifies an elapsed time value. ARF will begin monitoring the elapsed time of the job when CA-7 is notified of job initiation. If the elapsed time interval specified in the ARF definition expires prior to notification in CA-7 of job completion then an ARF exception is recognized. Such monitoring will allow CA-7 to warn of a possible problem if a job runs much longer than expected.

IS - Interrupt Submission

A test is made just prior to job submission similar to the test that is made for the LS condition. The ARF definition specifies a date, time and relational operator. If the date and time at job submission stands in the indicated relation to the date and time in the ARF definition, an ARF condition is recognized. ARF will then automatically requeue the job to the REQUEST queue prior to executing the responses.

JC - Job Completion Check

The execution status of a job is checked at job completion. Up to six completion code tests joined by Boolean operators may be specified in the ARF definition. Each test allows specification of the type of code to be tested: SYS for system abends, USR for user abends, CC for condition codes and FL for flush conditions. The test also allows a relational operator to be specified along with the value of the code. Thus, for example a test may specify 'CC GE 0024'. Only if the condition code of the job is greater than or equal to 24 is an ARF condition recognized. Masking for wild cards and generics is allowed. The following test may be specified as part of an ARF JC definition: SYS EQ +37. This indicates that any system abend ending in 37 such as S-D37 or S-E37 will be recognized as an ARF exception.

LA - Late Notification at CA-7 Prompting

An ARF exception is recognized when CA-7 begins late prompting for the job. The exception occurs only once for the job when it is initially considered late by CA-7. Also, the exception is only taken if the job becomes late while in the request queue, not when it enters the request queue as late. The prompt must actually be done. For example, if the JOB screen uses PROMPTS of N, the exception does not occur even though the job may show with a late status.

LB - Late Notification When Job Begins

A test is made when CA-7 is notified of job initiation. The ARF definition specifies a date, time and relational operator. If the date and time at job initiation stands in the relation specified to the date and time in the ARF definition, then an ARF exception is recognized.

LE - Late Notification When Job Ends

A test is made when CA-7 is notified of job completion. The ARF definition specifies a date, time and relational operator. If the date and time at job completion stands in the relation specified to the date and time in the ARF definition, then an ARF exception is recognized.

LS - Late Notification At Job Submission

A test is made just before CA-7 submits the job. The ARF definition specifies a date, time and relational operator. If the date and time at job submission stands in the relation specified to the date and time in the ARF definition, then an ARF exception is recognized.

SC - Step Completion Check

The execution status of a job is checked at step completion. Up to six completion code tests joined by Boolean operators may be specified in the ARF definition. Each test allows specification of the type of code to be tested: SYS for system abends, USR for user abends, CC for condition codes and FL for flush conditions. The test also allows a relational operator to be specified along with the value of the code. Thus, for example a test may specify 'CC GE 0024'. Only if the condition code of the job is greater than or equal to 24 is an ARF condition recognized. Masking for wild cards and generics is allowed. The following test may be specified as part of an ARF JC definition: SYS EQ +37. This indicates that any system abend ending in 37 such as S-D37 or S-E37 will be recognized as an ARF exception.

6.4 ARFSET Structure

Each production job that is to be monitored by ARF must specify the name of the set of ARF definitions that will supply the criteria that ARF uses to recognize and respond to exception conditions.

An ARFSET is a named collection of ARF definitions that can be referred to on the DB.1 panel, on a scheduled override statement in production JCL, or on a DEMAND command. If no ARFSET reference is provided when a job enters the request queue there will be no ARF monitoring for the job. The ARFSET specification on the DB.1 panel supplies the ARFSET reference unless it is overridden with the #ARF scheduled override card in the JCL. Both of these indications may be overridden on the DEMAND command by specifying the ARFSET keyword.

ARFSETs are created and maintained using the AR.3 panel. An ARFSET contains from 1 to 20 distinct definitions. When an ARFSET is created the client must supply a UID (like the UID specified on the DB.1 panel) along with a RESPONSE ID. The RESPONSE ID will be used to logon to an internal terminal where the responses will be executed when the exception is detected.

The naming conventions that are in effect for CA-7 job definition also apply to ARFSETs.

6.4.1 ARF Definition Structure

Each ARFSET contains from 1 to 20 definitions. Each definition in the set must be given a definition index that is unique within the ARFSET. The value of the definition index (or DEFID) can range from 1 to 255.

Each definition must also specify the type of exception condition that is being defined.

In addition to the definition index and type, the definition may be considered to have four parts: 1) the filter criteria, 2) the type specific tests, 3) the responses and 4) the final disposition.

6.4.2 Filter Criteria

The filter criteria determine whether the tests for a particular type of exception are applicable for this run of the job. If a job enters the request queue with an ARFSET reference, then the job is considered a candidate for ARF monitoring. The filter criteria are then used to determine whether or not the job is to be monitored by ARF.

The ARF definition contains criteria for tests of the system name, SCHID, restart count, entry mode, and queue entry date and time. The job entering the request queue must pass ALL of the tests implied by the filter criteria in order to be monitored by ARF.

6.4.3 Type Specific Tests

If a job passes all of the tests implied by the filter criteria and is monitored by ARF, then the type specific tests in the ARF definition will be applied at the point in processing implied by the exception type.

The type specific tests differ according to the type of exception condition that is to be defined. For example, the type specific tests for an SC (step completion check) include tests of the completion codes for the step. However, a different set of type specific tests must be applied to evaluate an LS (late notification at job submission) condition.

6.4.4 Responses

The definition of an ARF condition includes a set of actions that are to be executed in response to the exception. The responses are numbered in the ARF definition from 1 to 7. When an ARF exception is detected, the responses associated with the exception are queued for execution. If no other responses are executing for that job, the ARF dispatcher will begin executing the responses for the exception in the order that they appear in the ARF definition.

6.4.4.1 Types of Responses

Responses are defined using action statements. The format of an action statement is much like that of a CA-7 top line command. Each action statement begins with a two character code that declares the kind of action to be executed. Additional parameters are supplied with keywords and are delimited by commas.

A description of the four types of action statement follows. A more detailed discussion of the format and use of action statements will be found in the section devoted to the AR.3 panel.

AC - Issue a command

The AC action statement is used to issue a CA-7 command. The only parameter on the AC statement is the text of the command to be issued.

AW - Wait

Execution of the AW action statement causes the sequence of recovery actions to pause for a specified number of minutes.

AM - Issue a message

The AM action statement can be used to issue messages to the MVS console or to a specified TSO user.

AJ - Schedule a recovery job

The AJ action statement is used to schedule a job defined in the CA-7 database to run on behalf of the job that is in recovery. The AJ action statement is used to name the recovery job and to specify the number of times that the job is to be retried in the event that it does not complete successfully. A time interval to wait between such retries can also be coded on this statement.

Note: The AW and AJ action statements are only valid for use with the IS and JC exception conditions. Thus for exception types EC, EE, LA, LB, LE, LS and SC only the AC and AM action statements are allowed. All action statement types are valid for use with IS and JC exceptions.

6.4.5 Final Disposition

The final disposition indicates what action to take when all responses are complete for a job completion exception. If an IS or JC condition is to be defined, a final disposition must be specified. The final disposition is not valid for any other exception type.

Several options are available. The job may be restarted, it may be canceled, a normal completion may be forced or if no ARF action is desired, the job may be left for manual exception handling.

6.5 Implementation Considerations

Although every attempt has been made to ensure that resources are efficiently used by ARF, additional resources must be expended in order to monitor ARF exceptions. The impact of ARF on overall CA-7 throughput (if any) will vary depending on the extent of its implementation. It is therefore recommended that use of ARF be restricted to automating recovery for jobs with recurrent and predictable problems or for those jobs which are on a "critical path". If ARF is used to extensively monitor all jobs in the production workload, significant performance degradation is possible.

6.6 AR.3 ARF Condition Definition Maintenance Screen

Use this screen to create, update, delete, or review ARFSETs.

----- CA-7 ARF CONDITION DEFINITION MAINTENANCE -----
FUNCTION: (CLEAR,DELETE,EDIT,FE,FETCH,REPL,SAVE)

SET NAME:

UID :

RESPONSE ID:

DEF-COUNT: 000

PROGRAM: AR31 MSG-INDX: 00 -- AR.3 -- yy.ddd / hh:mm:ss
MESSAGE: ENTER FUNCTION, TRANSFER OR ENTER A COMMAND ON THE TOP LINE

To display, enter:

- AR.3 as the function value on any menu or formatted input screen.
- AR.3 as a top line command.

To exit:

- Move the cursor to the top line of the screen and enter the desired top line command.

6.6.1 Field Descriptions

FUNCTION:	The function to be performed. Value must be the name of some other screen or one of the following:
CLEAR	Clears the screen input data fields and resets the DEF-COUNT to 0.
DELETE	Deletes an ARFSET from the database.
EDIT	Transfers the user to the ARF definition editor where individual definitions within the ARFSET can be accessed.
FE	A combination of the FETCH and EDIT commands. Retrieves all definitions in the ARFSET and transfers to the ARF definition editor.
FETCH	Retrieves definitions for an ARFSET and puts them in the edit work area for subsequent access through the ARF definition editor.

REPL Replaces the existing ARFSET on the database.

SAVE Adds a new ARFSET to the database.

Required: Yes

Batch keyword: Positional parameter

SET NAME: The required name of the ARFSET that is being defined.

Size/Type: 1 to 8 alphanumeric characters

Required: Yes

Batch keyword: SET

UID: The required CA-7 user security identification.

Size/Type: 1 to 3 numeric characters from 0 to 255

Required: Yes

Batch keyword: UID

RESPONSE ID: The ID that will be used during ARF recovery. ARF uses an internal CA-7 terminal to process any commands that are needed for ARF recovery. This ID must have the authority to issue any commands that the ARF recovery sequence implies. For example, if the ARFSET contains a JC definition that has a final disposition of RESTART, then the RESPONSE ID will be used to issue a RESTART command for the job.

Size/Type: 1 to 8 alphanumeric characters (Maximum length allowable may vary depending on the external security package used, if any)

Required: Yes

Batch keyword: RESPONDR

DEF-COUNT: A display of the number of definitions that are in the ARF edit work file.

6.7 ARF Condition Definition Edit Screen

Use this screen to create, update, delete, or review individual definitions in an ARFSET.

```

----- ARF CONDITION EDIT -----
FUNCTION:      (ADD,DELETE,EXIT,FORMAT,LIST,REPL,SAVE,SR,SS)  DEFCT: 00
TYPE:          SYS EQ *      SID EQ 0      RSTC GE 0      EM EQ *      DEFID:
              FROM: 01011975 0001      TO: 12312074 2359

JC, SC TST: STEP EQ *      PROC EQ *      PGM EQ *
CC/ABENDS : ??? GE 0000  _  ??? GE 0000  _  ??? GE 0000  _
              ??? GE 0000  _  ??? GE 0000  _  ??? GE 0000  _
EC, EE, IS, LB, LE, LS TST:
RO: GE DATE:              TIME:              AO: ? INT/ADJ:

RESPONSES:
1:
2:
3:
4:
5:
6:
7:
FINAL -- DISP : N CA-11?: N BYPGDG: N USAGE:  PROCESS:  CC:
          START :              END :
PROGRAM: AR32 MSG-INDX: 00 -- AR.3.1 -- yy.ddd / hh:mm:ss
MESSAGE: ENTER FUNCTION, TRANSFER OR ENTER A COMMAND ON THE TOP LINE

```

To display, enter:

- EDIT as the function on the AR.3 screen.

To exit:

- Enter EXIT in the function field.
- Move the cursor to the top line of the screen and enter the desired top line command.

6.7.1 Field Descriptions

----- ARF CONDITION EDIT -----

FUNCTION:

(ADD,DELETE,EXIT,FORMAT,LIST,REPL,SAVE,SR,SS)

DEFCT: 00

TYPE:

SYS EQ *

SID EQ 0

RSTC GE 0

EM EQ *

DEFID:

FROM: 01011975 0001

TO: 12312074 2359

FUNCTION:

The function to be performed. Value must be the name of some other screen or one of the following:

- ADD** Adds a new definition to the work area.
- DELETE** Deletes a definition from the work area.
- EXIT** Returns the user to the AR.3 screen and clears the edit work area. Any changes are ignored. The DEF-COUNT has the value that it had before EDIT was entered.
- FORMAT** Clears any screen input entered by user.
- LIST** Lists the definition associated with the current definition index. This function may be used to browse definitions in the ARFSET.
- REPL** Replaces the current definition for this definition index in the work area. The definition must be LISTed before it can be REPLaced.
- SAVE** Updates the DEF-COUNT and returns to AR.3 panel. The ARF edit work area changes are retained but the ARFSET is not yet updated on the database.
- SR** Combination function which results in a SAVE, a return to the AR.3 screen, and a REPL from the AR.3 screen to replace an existing ARFSET on the database.
- SS** Combination function which results in a SAVE, a return to the AR.3 screen, and a SAVE from the AR.3 screen to add a new ARFSET to the database.

Required: Yes
Batch keyword: Positional parameter

DEFCT: Online display of the total number of ARF definitions in the ARF edit work area.

- TYPE:** Specifies the type of ARF exception condition to be defined. This parameter is required and the value must be one of the following:
- EC** Specifies an elapsed time check at normal job completion.
 - EE** Specifies an elapsed time check during job execution.
 - IS** Specifies a test of the date and time just prior to job submission. If an exception is detected, the job is requeued prior to executing ARF responses.
 - JC** Specifies a test for an exception condition based on completion codes to be executed when the job completes.

Note: Tests for job completion are made at each step completion. However, ARF does not respond to a job completion until the job ends. In the event that several job completion criteria are met, the responses associated with the lowest definition index are executed. Only one job completion condition occurs for a given job.
 - LA** Specifies an exception condition results when CA-7 initially prompts for a late condition. Note that no type specific tests are defined for the LA exception.
 - LB** Specifies a test to determine whether the job is to be regarded late when the job begins.
 - LE** Specifies a test to determine whether the job is to be regarded late when the job ends.
 - LS** Specifies a test to determine whether the job is to be regarded late when the job is submitted.
 - SC** Specifies a test for an exception condition based on completion codes to be executed when the specified step completes.
- Required: Yes
Batch keyword: TYPE
- DEFID:** An index which uniquely specifies this definition within the ARFSET.
- Size/Type: 1 to 3 decimal characters
Valid values 1-255
- Required: Yes
Batch keyword: DEFID

6.7.2 Field Descriptions - Filter Criteria

----- ARF CONDITION EDIT -----

FUNCTION:

(ADD,DELETE,EXIT,FORMAT,LIST,REPL,SAVE,SR,SS)

DEFCT: 00

TYPE:

SYS EQ *

SID EQ 0

RSTC GE 0

EM EQ *

DEFID:

FROM: 01011975 0001

TO: 12312074 2359

SYS

The following two input fields specify a test to be made on the system name associated with the job when the job enters the request queue to determine whether or not the job is to be monitored by ARF. Following the literal 'SYS' is a relational operator input field which is set by default to 'EQ' indicating that the system name of the job must be equal to the value in the following input field. This may be overtyped with an 'NE' to indicate that the system name must not be equal to the value in the following input field.

Required:

Yes

Default:

EQ

Batch keyword:

SYSRO

The input field immediately following the relational operator is set by default to '*'. The default values for SYS, 'EQ *' indicate that all system names are valid. The '*' may be changed to provide a fully or partially qualified system name to be compared against the system name of the job entering the request queue. The value may incorporate '+' to indicate a wild card. An '*' terminating the string indicates a generic. Thus for example, if the values for SYS are 'EQ PAY+23*', then any system name beginning with 'PAY' and having '23' in the fifth and sixth positions will pass this test.

Size/Type:

1 to 8 alphanumeric characters

+ indicates a wild card

* indicates a generic

Required:

Yes

Default:

*

Batch keyword:

SYS

SID

The following two input fields specify a test to be made on the SCHID associated with the job when the job enters the request queue to determine whether or not the job is to be monitored by ARF. Following the literal 'SID' is a relational operator input field which is set by default to 'EQ' indicating that the SCHID of the job must be equal to the value in the following input field. This may be overtyped with one of the following values:

EQ

equal

NE

not equal

GE

greater than or equal to

GT greater than
LE less than or equal to
LT less than

The SCHID of the job entering the request queue will be compared to the value in the input field following the relational operator. If the resulting statement is true then the test is passed.

Required: Yes
 Default: EQ
 Batch keyword: SIDRO

The input field immediately following the relational operator is set by default to '0'. The default values for SID, 'EQ 0' indicate that all SCHIDs are valid. The '0' may be changed to provide a fully qualified SCHID value from 001-255 to be compared against the SCHID of the job entering the request queue.

Size/Type: 1 to 3 decimal characters from 0 to 255
 Required: Yes
 Default: 0 (a value of 0 refers to ALL SCHIDs)
 Batch keyword: SID

RSTC

The following two input fields specify a test to be made on the restart count of the job to determine whether or not the job is to be monitored by ARF. Following the literal 'RSTC' is a relational operator input field which is set by default to 'GE' indicating that the restart count of the job must be greater than or equal to the value in the following input field. This may be overtyped with one of the following values:

NE not equal
GE greater than or equal to
EQ equal
GT greater than
LE less than or equal to
LT less than

The restart count of the job will be compared to the value in the input field following the relational operator. If the resulting statement is true then the test is passed.

Required: Yes
 Default: GE
 Batch keyword: RSTRO

The input field immediately following the relational operator is set by default to '0'. The default values for RSTC, 'GE 0' indicate that any restart count is valid. The '0' may be changed to provide a fully qualified decimal value from 0 to 999 to be compared against the restart count of the job to be considered for ARF monitoring.

Size/Type: 1 to 3 decimal characters from 0 to 999
 Required: Yes
 Default: 0
 Batch keyword: RST

EM

The following two input fields specify a test to be made on the entry mode of the job when the job enters the request queue to determine whether the job is to be monitored by ARF. Following the literal 'EM' is a relational operator input field which is set by default to 'EQ' indicating that the entry mode of the job must be equal to the value in the following input field. This may be overtyped with an 'NE' to indicate that the entry mode must not be equal to the value in the following input field.

Required: Yes
 Default: EQ
 Batch keyword: EMRO

The input field immediately following the relational operator is set by default to '*'. The default values for EM, 'EQ *' indicate that all entry modes are valid. The '*' may be changed to provide a fully qualified entry mode designation to be compared against the entry mode of the job entering the request queue. Entry mode designations valid in this field are:

DEMD Job entered through a DEMAND command

RUN Job entered through a RUN command

SSCN Job entered through a schedule scan

JTRG Job entered through a job trigger

DTRG Job entered through a DSN trigger

NTRG Job entered through a network trigger

PSCH Job entered through personal scheduling

Required: Yes
 Default: * (all entry modes)
 Batch keyword: EM

FROM

Two input fields used to specify a date and time that is compared against the date and time when the job enters the request queue. In order to pass the test, the queue entry date and time must be greater than the date and time specified here.

FROM Date

Size/Type: 8 numeric character positions in MMDDYYYY format. Valid range from 01011975 to 12312074. You can also use one of the reserved words in Table 6-1 on page 6-37.
 Required: Yes
 Default: 01011975
 Batch keyword: FROMD

FROM Time

Size/Type: 4 character positions in HHMM format. Valid range from 0000 to 2359.

Required: Yes

Default: 0001 (1 minute past midnight)

Batch keyword: FROMT

TO

Two input fields used to specify a date and time that is compared against the date and time when the job enters the request queue. To pass the test, the queue entry date and time must be less than the date and time specified here.

TO Date

Size/Type: 8 character positions in MMDDYYYY format. Valid range from 01011975 to 12312074. You can also use one of the reserved words in Table 6-1 on page 6-37.

Required: Yes

Default: 12312074

Batch keyword: TOD

TO Time

Size/Type: 4 character positions in HHMM format. Valid range from 0000 to 2359.

Required: Yes

Default: 2359 (11:59 PM)

Batch keyword: TOT

6.7.3 Field Descriptions - Type Specific Tests for JC and SC Conditions

JC, SC TST

JC, SC TST: STEP EQ *
CC/ABENDS : ??? GE 0000 _
EC, EE, LB, LE, LS TST:
RO: GE DATE: TIME: AO: ? INT/ADJ:

PROC EQ *
??? GE 0000 _
??? GE 0000 _

PGM EQ *
??? GE 0000 _
??? GE 0000 _

STEP The following two input fields specify a test to be made on the step name (or stepname within proc) at step or job completion for an ARF monitored job. Following the literal 'STEP' is a relational operator input field which is set by default to 'EQ' indicating that the step name of the job must be equal to the value in the following input field. This may be overtyped with an 'NE' to indicate that the step name must not be equal to the value in the following input field.

Required: Yes
Default: EQ
Batch keyword: STEPRO

The input field immediately following the relational operator is set by default to '*'. The default values for STEP, 'EQ *' indicate that all step names are valid. The '*' may be changed to provide a fully or partially qualified step name to be compared against the step name of the job being monitored by ARF. The value may incorporate '+' to indicate a wild card. An '*' terminating the string indicates a generic. Thus for example, if the values for STEP are 'EQ STEP++1*', then any step name beginning with 'STEP' and having '1' in the seventh position will be considered a candidate for an ARF exception.

Size/Type: 1 to 8 characters, blanks and commas are not allowed
+ indicates a wild card
* indicates a generic
Required: Yes
Default: *
Batch keyword: STEP

PROC The following two input fields specify a test to be made on the name of the step executing the proc at step or job completion for an ARF monitored job. Following the literal 'PROC' is a relational operator input field which is set by default to 'EQ' indicating that the proc name of the job must be equal to the value in the following input field. This may be overtyped with an 'NE' to indicate that the proc name must not be equal to the value in the following input field.

Required: Yes
 Default: EQ
 Batch keyword: PROCRO

The input field immediately following the relational operator is set by default to '*'. The default values for PROC, 'EQ *' indicate that all proc names are valid. The '*' may be changed to provide a fully or partially qualified proc name to be compared against the proc name of the job being monitored by ARF. The value may incorporate '+' to indicate a wild card. An '*' terminating the string indicates a generic. Thus for example, if the values for PROC are 'EQ PROC++1*', then any proc name beginning with 'PROC' and having '1' in the seventh position will be considered a candidate for an ARF exception.

Size/Type: 1 to 8 characters, commas and blanks are not allowed
 + indicates a wild card
 * indicates a generic

Required: Yes
 Default: *
 Batch keyword: PROC

PGM

The following two input fields specify a test to be made on the program name at step or job completion for an ARF monitored job. Following the literal 'PGM' is a relational operator input field which is set by default to 'EQ' indicating that the program name of the job must be equal to the value in the following input field. This may be overtyped with an 'NE' to indicate that the program name must not be equal to the value in the following input field.

Required: Yes
 Default: EQ
 Batch keyword: PGMRO

The input field immediately following the relational operator is set by default to '*'. The default values for PGM, 'EQ *' indicate that all program names are valid. The '*' may be changed to provide a fully or partially qualified program name to be compared against the program name of the job being monitored by ARF. The value may incorporate '+' to indicate a wild card. An '*' terminating the string indicates a generic. Thus for example, if the values for PGM are 'EQ PROG++1*', then any program name beginning with 'PROG' and having '1' in the seventh position will be considered a candidate for an ARF exception.

Size/Type: 1 to 8 characters, commas and blanks are not allowed
 + indicates a wild card
 * indicates a generic

Required: Yes
 Default: *
 Batch keyword: PGM

CC/ABENDS

Up to six tests of completion codes may be coded here and joined by Boolean operators to create complex completion code tests.

The first input field in a completion code test allows specification of the format of the completion code test. Overtyping this field with one of the following values in order to define a completion code test:

???	Indicates no completion code test.
SYS	The completion code to be tested is a system abend code.
USR	The completion code to be tested is a user abend code.
CC	The completion code to be tested is a condition code.
FL	A test for a flush condition is indicated. If this value is used, the corresponding completion code field is ignored.
JCL	A test for a JCL error is indicated. If this value is used, the corresponding completion code is ignored.

Required: No
 Default: ??? (indicates no test)
 Batch keyword: FMT1,FMT2,FMT3,FMT4,FMT5,FMT6

The second input field in a completion code test allows specification of the relational operator to be used in the completion code test. The default is 'GE'. This may be overtyped with one of the following values:

NE	not equal
GE	greater than or equal to
EQ	equal
GT	greater than
LE	less than or equal to
LT	less than

Required: No, unless the first input has a value other than ???
 Default: GE
 Batch keyword: RO1,RO2,RO3,RO4,RO5,RO6

The third input field in a completion code test allows specification of the completion code value to be used in a completion code test. The default is '000'. This value may be overtyped with a character value appropriate for the format of the completion code test. For example, if the first input field in the test contains 'SYS' indicating a system abend, then a valid completion code value might be 'E37'. The completion code value may contain '+' to indicate a wild card and '*' for generics. For example, the following completion code test would cause an ARF exception to be recognized for any system abend ending in '37': 'SYS EQ +37'.

Size/Type: For SYS, 3 valid hexadecimal characters
 For USR and CC, 1 to 4 decimal characters
 Required: No, unless the first input has a value other than ???
 Default: 000
 Batch keyword: VAL1,VAL2,VAL3,VAL4,VAL5,VAL6

Completion code tests may be joined using the following logical connectives:

& Indicates the tests are to be 'and'ed

/ Indicates the tests are to be 'or'ed

Required: No, unless tests are to be joined
 Default: __ indicating no connective
 Batch keyword: LO1,LO2,LO3,LO4,LO5

6.7.4 Field Descriptions - Type Specific Tests for EC, EE, IS, LB, LE and LS Conditions

JC, SC TST: STEP EQ *
PROC EQ *
PGM EQ *

CC/ABENDS : ??? GE 0000 _
??? GE 0000 _
??? GE 0000 _

EC, EE, IS, LB, LE, LS TST:
RO: GE DATE:
TIME:
AO: ? INT/ADJ:

The following fields are used to define condition tests for EC, EE, IS, LB, LE and LS condition types. The fields required vary according to the condition type to be defined.

- RO:

This input field may be used with the following condition types: EC, IS, LB, LE and LS. Data from this field as well as the TIME, AO and INT/ADJ fields is used to construct a date and time test to determine if an ARF condition exists. The relational operator for the test is provided in this field.

If the condition type is IS, the date/time when CA-7 attempts to submit the job is compared with the date/time value built from the input fields following this relational operator. An ARF condition occurs if the indicated relation obtains.

If the condition type is LB, the date/time when the job begins is compared with the date/time value that is built from data provided in the input fields following this relational operator. An ARF condition occurs if the indicated relation obtains.

If the condition type is LE, the date/time when the job ends is compared with the date/time expression that is built from data provided in the input fields following this relational operator. An ARF condition occurs if the indicated relation obtains.

If the condition type is LS, the date/time when the job submits is compared with the date/time expression that is built from data provided in the input fields following this relational operator. An ARF condition occurs if the indicated relation obtains.

If the condition type is EC, the elapsed time when the job completes is compared with the value in the input field tagged INT/ADJ. An ARF condition occurs if the indicated relation obtains.

The relational operator is ignored for the EE test. The only valid input field is INT/ADJ.

The default is 'GE'.

This may be overtyped with one of the following values:

NE not equal

GE greater than or equal to

GT greater than

EQ equal

LE less than or equal to

LT less than

Required: Required for EC, IS, LB, LE and LS condition types. It should not be specified for any other condition type.

Default: GE

Batch keyword: TRO

DATE: This input field may be used with the following condition types: IS, LB, LE and LS. Data from this field as well as the TIME, AO, and INT/ADJ fields is used to construct a date and time that is used in a comparison to determine if an ARF condition exists. An ARF condition occurs if the relation indicated by the RO value obtains.

If the condition type is IS, the date/time when CA-7 attempts to submit the job is compared with the date/time expression specified here.

If the condition type is LB, the date/time when the job begins is compared with the date/time expression specified here.

If the condition type is LE, the date/time when the job ends is compared with the date/time expression that is specified here.

If the condition type is LS, the date/time when the job submits is compared with the date/time expression that is specified here.

Size/Type: 8 character positions in MMDDYYYY format. Valid range from 01011975 to 12312074. You can also use one of the reserved words in Table 6-1 on page 6-37.

Required: Required for IS, LB, LE and LS condition types. It should not be specified for any other condition types.

Batch keyword: TDAT

TIME: This input field may be used with the following condition types: IS, LB, LE and LS. Data from this field as well as the DATE, AO and INT/ADJ fields is used to construct a date and time that is used in a comparison to determine if an ARF condition exists.

If the condition type is IS, the date/time when CA-7 attempts to submit the job is compared with the date/time expression specified here.

If the condition type is LB, the date/time when the job begins is compared with the date/time expression specified here.

If the condition type is LE, the date/time when the job ends is compared with the date/time expression that is specified here.

If the condition type is LS, the date/time when the job submits is compared with the date/time expression that is specified here.

Size/Type: 4 character positions in HHMM format. Valid range from 0000 to 2359. You can also use one of the reserved words in Table 6-1 on page 6-37.

Required: Required for IS, LB, LE and LS condition types. It should not be specified for any other condition types.

Batch keyword: TTIM

AO: This input field may be used with the following condition types: IS, LB, LE and LS. Data from this field as well as the DATE, TIME and INT/ADJ fields is used to construct a date and time that is used in a comparison to determine if an ARF condition exists.

Data from this field along with data in the INT/ADJ field is used to adjust the date/time value provided in the DATE and TIME fields. If the increment used as an adjustment is to be added, use '+', if it is to be subtracted, use '-'.

If the condition type is IS, the date/time when CA-7 attempts to submit the job is compared with the date/time expression specified here.

If the condition type is LB, the date/time when the job begins is compared with the date/time expression specified here.

If the condition type is LE, the date/time when the job ends is compared with the date/time expression that is specified here.

If the condition type is LS, the date/time when the job submits is compared with the date/time expression that is specified here.

Size/Type: 1 position. Allowable values are '+' or '-'.

Required: Only allowable on IS, LB, LE and LS condition types. It should not be specified for any other condition types. It should only be used if an adjustment value is provided in the INT/ADJ field.

Batch keyword: TAO

INT/ADJ: This input field may be used with the following condition types: EC, EE, IS, LB, LE and LS. Data from this field may be interpreted as an adjustment to a date and time already specified; this is its meaning if the condition type is LB, LE or LS. If the condition type is EC or EE, then the value in this field is interpreted as an elapsed time interval.

The value of this field should be specified in HHMM format, and is valid within ± 1 minute.

Size/Type: 4 character positions in HHMM format. Valid range from 0000 to 2359. You can also use one of the reserved words in Table 6-1 on page 6-37.

Required: Required for EE and EC
Allowed for IS, LB, LE, and LS.
It should not be specified for any other condition types.

Batch keyword: INT

6.7.5 Field Descriptions - Responses

RESPONSES:			
1:			
2:			
3:			
4:			
5:			
6:			
7:			
FINAL --	DISP	: N CA-11?: N BYPGDG: N	USAGE: PROCESS: CC:

In this section of the AR.3 screen there are seven lines where ARF responses may be specified. Responses may not be continued across lines. Each line of input is taken to be a separate response that will be executed when this ARF condition is detected.

A valid ARF action statement must be coded on each non-blank response line. Rules for coding ARF action statements are detailed in 6.8, "Rules for Coding ARF Action Statements" on page 6-31.

Size/Type: 74 character positions for input. Format must conform to rules documented in 6.8, "Rules for Coding ARF Action Statements" on page 6-31.

Required: No

Batch keyword: RSP1, RSP2, RSP3, RSP4, RSP5, RSP6, RSP7

Note: Batch syntax requires that action statements be enclosed in parentheses. Additional parentheses may be needed depending on action statement keywords and parameters.

6.7.6 Field Descriptions - Final Disposition

```

FINAL -- DISP : N CA-11?: N BYPGDG: N USAGE:  PROCESS:  CC:
        START :                      END :
PROGRAM: AR32  MSG-INDX: 00  -- AR.3.1  -- yy.ddd / hh:mm:ss
MESSAGE: ENTER FUNCTION, TRANSFER OR ENTER A COMMAND ON THE TOP LINE

```

This section of the AR.3 screen describes the action that is to be taken after all ARF responses have been issued.

Note: This section is ignored unless the condition type is IS or JC.

FINAL -- DISP: This input field indicates the action to be taken when all ARF responses have been issued. The valid values are:

- N** Take no action, the job is left for manual recovery. Subsequent restarts will continue to be monitored by ARF. 'N' is the default.
- R** Restart the job using the CA-7 top-line RESTART command. Additional parms for the command may be supplied using other input fields in the final disposition section. The JCL override requirement will always be posted if this option is used. This is not valid for an IS condition.
- C** Cancel the job using the CA-7 top-line CANCEL command.
- F** Force completion of this job using the RESTART command with the FORCECOMP option.
- P** Issue an ARFP command to purge ARF records for this job. Subsequent restarts will **not** be monitored by ARF.

Required: Yes if condition type is IS or JC. Ignored for all other condition types.

Default: N (no action)

Batch keyword: DSP

- CA-11?:** This input field indicates whether CA-11 is to be used for restart. The valid values are:
- N** No, CA-11 is not to be used for restart. This is the default.
 - Y** Yes, CA-11 is to be used for restart. Other input values in the final disposition section may be relevant.
- Required: Required only if CA-11 is to be used for restart.
 Default: N (CA-11 not used)
 Batch keyword: CA11
- BYPGDG:** This input field indicates the value on the RESTART command. See the documentation on the RESTART command for further details.
- Valid values are:
- C** CA-11 to accept GDG bias resolution according to the MVS catalog (CAT option of CA-11).
 - N** This is the default. CA-11 should NOT bypass GDG logic on restart.
 - V** CA-11 to verify that the GDG bias resolution recorded in the CMT agrees with the MVS catalog (VER option of CA-11).
 - Y** Yes, CA-11 should bypass GDG logic on restart.
- Required: NO
 Default: N (CA-11 will not bypass GDG logic)
 Batch keyword: BYPGDG
- USAGE:** Used to supply the value of the USAGE parameter to be used on the RESTART command. See CA-11 documentation for values. This option honored only if CA-11 used and CA-7 is inserting the RMS step. See the documentation on the INSERT-RMS field of the DB.1 screen and on the RESTART command for further details.
- Size/Type: 1 alphanumeric character
 Required: No
 Batch keyword: USAGE
- PROCESS:** Used to supply the value of the PROCESS parameter to be used on the RESTART command. This option honored only if CA-11 used and CA-7 is inserting the RMS step. Valid values are F, P, S, N, O or R. See the documentation on the INSERT-RMS field of the DB.1 screen and on the RESTART command for further details.
- Size/Type: 1 alphanumeric character
 Required: No
 Batch keyword: PROCESS

CC

Used to supply the value of the CONDCD parameter to be used on the RESTART command. This option honored only if CA-11 used and CA-7 is inserting the RMS step. See the documentation on the INSERT-RMS field of the DB.1 screen and on the RESTART command for further details.

Size/Type: 1 to 4 alphanumeric characters from 0-4095

Required: No

Batch keyword: CC

START:

Used to supply values of the PROCSTRT and STPSTRT parameters on the RESTART command. This option honored only if CA-11 used.

The format of the START value is 'procstep.step' where 'procstep' is the value of the PROCSTRT keyword and 'step' is the value of the STPSTRT keyword. If the value of this field does not contain a '.' then it is assumed that the input applies only to the STPSTRT keyword. If the step number is used, then PROCSTRT cannot be used.

See the documentation on the RESTART command for further details.

Size/Type: 1 to 17 alphanumeric characters. If procstep reference is provided, a '.' must separate procstep name and step name. You should validate this value prior to use to ensure that the steps named here are used by the job that uses this ARFSET.

Required: No

Batch keyword: START

END:

Used to supply values of the PROCEND and STPEND parameters on the RESTART command. This option honored only if CA-11 used.

The format of the END value is 'procstep.step' where 'procstep' is the value of the PROCEND keyword and 'step' is the value of the STPEND keyword. If the value of this field does not contain a '.' then it is assumed that the input applies only to the STPEND keyword. If the step number is used, then PROCEND cannot be used.

See the documentation on the RESTART command for further details.

Size/Type: 1 to 17 alphanumeric characters. If procstep reference is provided, a '.' must separate procstep name and step name. You should validate this value prior to use to ensure that the steps named here are used by the job that uses this ARFSET.

Required: No

Batch keyword: END

6.8 Rules for Coding ARF Action Statements

An ARF action statement is coded in the RESPONSE section of an ARF condition definition and describes an action or set of actions that will be taken in response to the occurrence of an ARF condition.

Each ARF statement begins with an identifier that indicates the type of action statement that is being coded. Parameters on an ARF action statement are provided in keyword format and are separated by commas.

There are four ARF action statement types. A description of each statement type and its associated parameters follows.

Note: The AW and AJ action statements are only valid for use with the IS or JC exception conditions. Thus for exception types EC, EE, LA, LB, LE, LS and SC only the AC and AM action statements are allowed. All action statement types are valid for use with IS and JC exceptions.

6.8.1 AC - Issue a Command

Statement identifier: AC

Purpose: The AC action statement is used to issue a CA-7 command.

Format: The allowable keyword is:

M= The data following the M= keyword is assumed to be valid CA-7 terminal input. The format of the command must adhere to the conventions used for the TRAILER terminal. The maximum length is 69.

6.8.1.1 Example

```
AC,M=DEMAND,JOB=PAYROLL
```

The example above illustrates the use of the AC action statement to issue a DEMAND command to request the PAYROLL job.

6.8.2 AM - Issue a Message

Statement identifier: AM

Purpose: Execution of the AM action statement causes messages to be issued to the MVS console or to a specified TSO user.

Format: The allowable keywords are:

- CM=** The data following the CM= specifies the communications mode for the message. The following are the acceptable values of CM=:
- T** Send the messages to the user whose TSO ID is specified using the U= keyword. The message is not retained if the specified TSO user is not logged on.
 - L** Send the messages to the user whose TSO ID is specified using the U= keyword. If the specified TSO user is not logged on, the message will be retained and may be displayed in the next TSO session.
 - C** Send the messages to the MVS console.
 - H** Send the messages to the MVS console and highlight them.
- U=** The data following the U= names the TSO user that is to receive the messages. This value is required if CM=T or CM=L. It is invalid for any other value of CM. This keyword can also be coded as USER=.
- M=** The data following the M= specifies the text of a message that is to be issued. If the text contains commas or keywords (x=y), enclose the entire text string in parentheses. The maximum length is 60 characters.

6.8.2.1 Examples

```
AM,CM=T,U=TSOUSER1,M=HELLO THERE!
```

The example above illustrates the use of the AM action statement to send the message "HELLO THERE!" to a TSO user.

```
AM,CM=C,M=HELLO CONSOLE!
```

The example above illustrates the use of the AM action statement to send the message "HELLO CONSOLE!" to the MVS console.

6.8.3 AW - Wait

Statement identifier: AW

Purpose: Execution of the AW action statement causes the sequence of recovery actions to pause for a specified number of minutes.

Format: The allowable keyword is:

TIME= The data following the TIME= keyword specifies the number of minutes that ARF should wait before proceeding to execute the next recovery action. This is a decimal number of minutes from 0000 to 9999. This keyword can also be coded as T=.

6.8.3.1 Example

```
AW,TIME=0002
```

The example above illustrates the use of the AW action statement to cause a 2 minute pause in ARF recovery for a given job.

6.8.4 AJ - Schedule a Recovery Job

Statement identifier: AJ

Purpose:

The AJ action statement is used to schedule a job (known as an 'ARFJ' job) to run on behalf of the job that is in recovery. ARF will not proceed to the next ARF action statement until this action statement is considered complete. An AJ action statement is complete when the ARFJ job completes successfully or when the specified number of retries is exhausted.

Note: The completion of an ARFJ job does not satisfy requirements, nor does it cause triggering of other jobs.

However, CA-7 tracks the progress of the ARFJ job and records its completion status in the run log and in messages to the master station.

Also, ARFSET designations (such as those on the DB.1 panel or on #ARF statements in the JCL) are ignored for ARFJ jobs. Thus, ARFJ jobs are not monitored for ARF recovery.

Format:

The allowable keywords are:

- JOB=** The value of this keyword names the job that is to run on behalf of the job that is in recovery. This keyword is required and there is no default.
- RETRY=** If the ARFJ job does not complete successfully, ARF can request the job again. The value of the RETRY parameter is the maximum number of times the job can be retried. The acceptable values are 0-9. The default value is RETRY=0.
- DELAY=** If the ARFJ job does not complete successfully, ARF can request the job again. The value on the DELAY keyword specifies the number of minutes before retrying. This value must be specified as a decimal number of minutes from 0 to 9999. The default value is DELAY=0000.

ACTION=

If the ARFJ job does not complete successfully and if all retries are exhausted, then the ACTION keyword value determines the disposition of the ARFJ job.

This keyword is required and there is no default value.

Acceptable values are:

E Suspend ARF recovery for this job. The LARFQ display indicates that the job is in *error* status.

N Take no action. Continue with next ARF action statement in sequence.

1-7 Jump to the ARF recovery action indicated.

6.8.4.1 Example

```
AJ,JOB=X,RETRY=1,DELAY=1,ACTION=4
```

The example above illustrates the use of the AJ action statement to request an ARFJ job named X. If the job does not complete successfully then retry after waiting 1 minute. If the job still does not complete successfully jump to the 4th action statement in the definition.

6.9 Using Reserved Words in Type Specific Tests

Certain values are not capable of being exactly defined at the time an ARF condition is defined. For example, the due-out time may not be known until the job enters the request queue. For such values, ARF provides a limited number of reserved words that can be used in certain fields on the AR.3.1 screen. The value of an ARF reserved word is resolved at the time the value is needed for ARF condition evaluation.

The following table describes each of these reserved words:

Table 6-1. Reserved Words		
Name	Description	Relevant AR.3 screen fields
DOD	The due-out date of the job.	FROM: (first input field) TO: (first input field) DATE:
DOT	The due-out time of the job.	FROM: (second input field) TO: (second input field) TIME:
DLD	The deadline date of the job.	FROM: (first input field) TO: (first input field) DATE:
DLT	The deadline time of the job.	FROM: (second input field) TO: (second input field) TIME:
EL	The current value from JQJELTME for this job; the elapsed time used by the job.	INT/ADJ:

These reserved words may be used to build arithmetic expressions as in the following example:

DLD+12

This expression is taken to indicate the deadline date plus 12 days.

Syntax: If used in an arithmetic expression, the reserved word must be followed immediately (no spaces) by an arithmetic operator (either + or -). If an arithmetic operator is present, it must be followed immediately (no spaces) by a decimal numeric value (up to 4 digits). The value of the reserved word will be incremented or decremented by the amount specified. The unit of the increment/decrement depends on the reserved word. For example, if the expression is DOT+0020, this is interpreted as the due-out time plus 20 minutes.

6.10 Using Variables in ARF Action Statements

Certain values in ARF action statements may be referenced using variables. These variables can be used in AM and AC action statements. The following table describes each of these variables:

Name	Description
&ARFSET	The name of the ARFSET referenced by this run of the job.
&AARDATE	The date the ARFSET referenced by this job was first used for ARF monitoring.
&AARTIME	The time the ARFSET referenced by this job was first used for ARF monitoring.
&ARFRESP#	The number of the response being processed.
&ARFSETIX	The index of the definition being processed.
&INTDATE	The date this ARF condition was detected.
&INTTIME	The time this ARF condition was detected.
&JOB#	The CA-7 job number of the job in ARF recovery.
&JOBNAME	The name of the job in ARF recovery.

6.11 Examples of ARF Condition Definition

The following pages contain detailed examples of using the AR.3 panels to define ARF conditions and responses. Additional examples are available through the online HELP facility. An index of ARF help topics can be found in the AR00 help member.

6.11.1 Defining a Job Completion Condition (JC)

Suppose that the PAYROLL job is known to suffer repeated x37 system abends. The response to an x37 abend for the PAYROLL job is to notify the operator that the REORG job is to be run. When the REORG job completes, the PAYROLL job should be restarted. The job may be restarted exactly one time.

This example illustrates how CA-7 ARF may be used to automate this recovery procedure. The definition is coded as in the following figure:

```

----- ARF CONDITION EDIT -----
FUNCTION: ADD      (ADD,DELETE,EXIT,FORMAT,LIST,REPL,SAVE,SR,SS) DEFCT: 00
TYPE: JC      SYS EQ *      SID EQ 0      RSTC EQ 0      EM EQ *      DEFID: 1
          FROM: 01011975 0001      TO: 12312074 2359

JC, SC TST: STEP EQ *      PROC EQ *      PGM EQ *
CC/ABENDS : SYS EQ +37    ___   ??? GE 0000    ___   ??? GE 0000    ___
          ??? GE 0000    ___   ??? GE 0000    ___   ??? GE 0000    ___
EC, EE, IS, LB, LE, LS TST:
          RO: GE DATE:      TIME:      AO: ? INT/ADJ:

RESPONSES:
1: AM,CM=H,M=ATTENTION! &JOBNAME(&JOB#) HAS ABENDED!
2: AJ,JOB=REORG,DELAY=0000,RETRY=0000,ACTION=N
3: AM,CM=H,M=ATTENTION! &ARFSET ACTIONS COMPLETE FOR &JOBNAME(&JOB#)
4:
5:
6:
7:
FINAL -- DISP : R CA-11?: N BYPGDG: N USAGE: PROCESS: CC:
          START :      END :
PROGRAM: AR32 MSG-INDX: 00 -- AR.3.1 -- yy.ddd / hh:mm:ss
MESSAGE: ENTER FUNCTION, TRANSFER OR ENTER A COMMAND ON THE TOP LINE

```

In this example note that the definition is new and is being added. Thus, the value in the FUNCTION field is 'ADD'. TYPE is required. The value here is 'JC' to indicate that a job completion condition is being defined. The definition requires an index, the value here is '1'.

Most of the defaults for the filter criteria are acceptable. However, this definition will only apply to the first run of the job. If it is restarted, this definition will not be applicable because the RSTC test specifies 'EQ 0'.

The type specific test indicates that it is to apply to any step, proc and program in the job. A system abend code test is indicated so that any 'x37' system abend is to be recognized as an ARF exception. ARF allows a wildcard specification as in '+37' to indicate that this definition is for any 'x37' abend.

Assume that a run of PAYROLL (CA-7 job number: 0211) has abended with a S-D37 abend. Assume too, that the definition in this example is used for ARF monitoring and is contained in ARFSET: PAYMON. When the condition is detected, ARF will send the following highlighted message to the MVS console:

ATTENTION! PAYROLL(211) HAS ABENDED!

Note the use of ARF variables to refer to the job name and CA-7 job number. After issuing this message, ARF will submit a job named REORG to run on behalf of PAYROLL. When the job completes, the following highlighted message will be sent to the MVS console:

ATTENTION! PAYMON ACTIONS COMPLETE FOR PAYROLL(211)

Because the condition defined is for job completion, the final disposition is processed when all other action statements for the definition have been handled. In this case, the job is to be restarted without CA-11.

6.11.2 Defining a Late at Job End Notification (LE)

The PAYROLL job has problems with x37 system abends, but it also has a tendency to run rather late on occasion. So ARF will be used to set up a notification based upon a test of an exact date and time. This test will be executed when the job completes. The notification should occur when the job completes if it is 30 minutes or more past the due-out date and time.

This example illustrates how CA-7 ARF may be used to automate this recovery procedure. The definition is coded as in the following figure:

```

----- ARF CONDITION EDIT -----
FUNCTION: ADD      (ADD,DELETE,EXIT,FORMAT,LIST,REPL,SAVE,SR,SS) DEFCT: 01
TYPE: LE   SYS EQ *      SID EQ 0   RSTC EQ 0   EM EQ *   DEFID: 2
          FROM: 01011975 0001      TO: 12312074 2359

JC, SC TST: STEP EQ *      PROC EQ *      PGM EQ *
CC/ABENDS : ??? GE 0000   ___  ??? GE 0000   ___  ??? GE 0000   ___
          ??? GE 0000   ___  ??? GE 0000   ___  ??? GE 0000   ___
EC, EE, IS, LB, LE, LS TST:
          RO: GE  DATE: DOD      TIME: DOT      AO: +  INT/ADJ: 0030
RESPONSES:
1: AM,CM=T,U=FRED,M=HEY FRED! YOUR PAYCHECK COULD BE DELAYED!!
2:
3:
4:
5:
6:
7:
FINAL -- DISP : N CA-11?: N BYPGDG: N  USAGE:  PROCESS:  CC:
          START :                      END :
PROGRAM: AR32  MSG-INDX: 00  -- AR.3.1  --  yy.ddd / hh:mm:ss
MESSAGE: ENTER FUNCTION, TRANSFER OR ENTER A COMMAND ON THE TOP LINE

```

In this example note that the definition is new and is being added. Thus, the value in the FUNCTION field is 'ADD'. The value of DEFCT is 1 which reflects the fact that a definition has already been added for the job completion condition in the example above. The TYPE value is required, the value here is 'LE' indicating a "late at job end" notification is to be defined. The definition also requires an index, the value here is '2'.

The defaults for the filter criteria are acceptable.

In the relevant type specific test section, the RO value is 'GE', the DATE is 'DOD', the TIME is 'DOT', the arithmetic operator is '+' and the INT/ADJ value is '0030'. When the job completes, ARF will test the date and time at job completion to determine if it is greater than or equal to the due-out date and time plus 30 minutes.

When the ARF condition is detected, the following message will be sent to the TSO user whose ID is 'FRED':

HEY FRED! YOUR PAYCHECK COULD BE DELAYED!!!

There are no other actions following this one. This is not a job completion event so there will be no final disposition processing. ARF recovery is complete for this exception.

Chapter 7. Workload Documentation

CA-7 permits the entry of free-form user documentation (prose) of the workload at virtually any level. This may range from a general description to specific instructions on how a particular function at a specified workstation is to be performed. Some of the documentation features are:

- Ready access to documentation through CA-7 terminals.
- Realtime, online updating from a change control or production control area.
- Segmenting documentation into user-defined categories (that is, restart information, user abends, control card layouts, balancing instructions, setup requirements, and so forth).
- Adding documentation (prose) through any authorized CA-7 terminal using the DB.4 screens and edit facility of database maintenance.

Once defined, you may define documentation:

- With the LPROS command.
- With the DB.4 screens (on which the documentation is also defined and maintained).
- As a list option in several other inquiry commands such as LQ, LPRE, LJOB, and so forth.
- Automatically in some cases. (See 7.2, “Special Documentation Features” on page 7-26.)

7.1 Workload Documentation Screens

There are six predefined major categories of documentation, each with its own unique definition screen. Within the documentation defined through these screens, the user may embed special user-defined segments as discussed in 7.2, “Special Documentation Features” on page 7-26. You may select the appropriate screen from the DB.4 Menu display. Separate screens are provided for:

- Job documentation
- System documentation
- Network documentation
- User-designated documentation
- Data set documentation
- DD documentation

If user documentation already exists, a user program could be coded to produce Batch Terminal Interface PROSE commands. These commands with the documentation could be the input to the Batch Terminal Interface program and the documentation could be added to the CA-7 database.

Also, if the documentation exists as members of a PDS that is available to CA-7, it could be fetched and edited using the DB.7 screen and saved in the Active Area. The user could then enter the top line DB.4 command, select the appropriate category, perform the EDIT function, and then SAVE the documentation to the database. Both of these methods could save the user from having to retype a long documentation description.

7.1.1 DB.4 Menu Screen

Use this screen for selecting various documentation functions.

```

----- CA-7 WORKLOAD DOCUMENTATION MENU -----
FUNCTION ==> xxxxxxxx

DOCUMENTATION FOR:
  1 - CPU JOB
  2 - INPUT/OUTPUT NETWORK
  3 - USER-DEFINED ITEM
  4 - DATA SET
  5 - DD STATEMENT
  6 - APPLICATION SYSTEM

PROGRAM: SM10 MSG-INDX: nn  -- DB.4      -- yy.ddd / hh:mm:ss
MESSAGE:

```

To display, enter:

- 4 as the FUNCTION on the DB Menu screen.
- DB.4 as the FUNCTION on any other menu or formatted input screen.
- DB.4 or PROSE as a top line command.

To exit:

- Press the PF3 key to return to the DB Menu screen except with the TSO/ISPF interface when PF3 has been assigned to some other function.
- Move the cursor to the top line of the screen and enter the desired top line command if yet another function is desired.

7.1.1.1 Usage Notes

The desired documentation type is selected by entering the appropriate FUNCTION value and pressing the Enter key.

7.1.2 DB.4.1 - CPU Job Documentation Screen

Use this screen for defining free-form job level documentation. Documentation may be defined in one continuous stream for the specified job. Also, it may be subdivided into segments and subsegments as discussed in 7.2, “Special Documentation Features” on page 7-26.

```
----- CA-7 CPU JOB DOCUMENTATION -----  
FUNCTION: xxxxxxxx (APPEND,CLEAR,DELETE,EDIT,FE,FETCH,LIST,REPL,SAVE,UPD)  
  
JOB: xxxxxxxx  
SYSTEM: xxxxxxxx  
  
DESC: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx LINK: nnnnnn  
  
ACTIVE SIZE: nnnn  
  
PROGRAM: SM11 MSG-INDX: nn -- DB.4.1 -- yy.ddd / hh:mm:ss  
MESSAGE:
```

To display, enter:

- 1 as the FUNCTION on the DB.4 Menu screen.
- 4.1 as the FUNCTION on the DB Menu screen.
- DB.4.1 as the FUNCTION on any other menu or formatted input screen.
- DB.4.1 or PROSEJOB as a top line command.

To exit:

- Press the PF3 key to return to the DB.4 Menu screen except with the TSO/ISPF interface when PF3 has been assigned to some other function.
- Move the cursor to the top line of the screen and enter the desired top line command if yet another function is desired.

7.1.2.1 Field Descriptions

FUNCTION:	<p>The function to be performed. Value must be the name of some other screen or one of the following:</p> <p>APPEND Retrieves documentation data from the database and attaches documentation text to the end of any existing lines in the Active Area (text work area).</p> <p>CLEAR Clears screen input data fields and clears the user's Active Area.</p> <p>DELETE Deletes a documentation member from the database.</p> <p>EDIT Transfers the user to the edit facility and allows text processing. See Chapter 11, "Edit Facility" on page 11-1 and following subtopics for a discussion of the available commands for processing the desired text.</p> <p>FE Combination FETCH and EDIT.</p> <p>FETCH Retrieves documentation data from the database and replaces the user's Active Area with documentation text.</p> <p>LIST Lists documentation data only. In batch, a formatted screen is not listed; only a found or not found message is returned.</p> <p>REPL Replaces a documentation member in the database.</p> <p>SAVE Adds a new documentation member to the database.</p> <p>UPD Updates DESC and LINK fields only. It does not affect the documentation text.</p> <p>Required: Yes Batch keyword: Positional parameter</p>
n/a	<p>This field has a constant value of JOB for job documentation.</p> <p>Required: Yes for batch only Batch keyword: Positional parameter</p>
JOB:	<p>The job name for which the indicated function is to be performed.</p> <p>Size/Type: 1 to 8 alphanumeric characters Required: Yes Batch keyword: JOB</p>
SYSTEM:	<p>Identifies the system name or system ID related to the job for the documentation being defined.</p> <p>Size/Type: 1 to 8 alphanumeric characters Required: No Batch keyword: SYSTEM</p>

DESC: An optional description of the job.

Size/Type: 1 to 45 alphanumeric characters

Required: No

Batch keyword: DESC

LINK: The number of another documentation member to link to the documentation member listed. Linking documentation members together causes them to be listed together as if they were a single member. The LINK member is listed after the requested member when using the LPROS command only.

Size/Type: 1 to 6 numeric characters

Required: No

Batch keyword: LINK

Note: It is possible to create an LPROS loop. Linked documentation should not be linked to itself. The following example shows a loop:

A linked to B
B linked to C
C linked to A

ACTIVE SIZE: A system-generated field which tells how many lines of text exist in the Active Area for the current terminal session.

7.1.3 DB.4.2 - Input/Output Network Documentation Screen

Use this screen for defining network level documentation. It should not be used if the SCHD-PROSE field on the DB.5 screen was used for the network. Documentation may be defined in one continuous stream for the specified network. Also, it may be subdivided into segments and subsegments as discussed in 7.2, “Special Documentation Features” on page 7-26.

```

----- CA-7 INPUT/OUTPUT NETWORK DOCUMENTATION -----
FUNCTION: xxxxxxxx (APPEND,CLEAR,DELETE,EDIT,FE,FETCH,LIST,REPL,SAVE,UPD)

JOB: xxxxxxxx
SYSTEM: xxxxxxxx

NETWORK: xxxxxxxx

DESC: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx LINK: nnnnnn

STEP: xxxxxxxx DDNAME: xxxxxxxx
REPORT-ID: xxxxxxxxxxxxxxxxxxxxxxxx
FORM: xxxxxxxx TRAIN: xx CARRIAGE: xxxx COPIES: nnn

ACTIVE SIZE: nnnn

PROGRAM: SM11 MSG-INDX: nn -- DB.4.2 -- yy.ddd / hh:mm:ss
MESSAGE:

```

To display, enter:

- 2 as the FUNCTION on the DB.4 Menu screen.
- 4.2 as the FUNCTION on the DB Menu screen.
- DB.4.2 as the FUNCTION on any other menu or formatted input screen.
- DB.4.2 or PROSE,NWK as a top line command.

To exit:

- Press the PF3 key to return to the DB.4 Menu screen except with the TSO/ISPF interface when PF3 has been assigned to some other function.
- Move the cursor to the top line of the screen and enter the desired top line command if yet another function is desired.

7.1.3.1 Field Descriptions

FUNCTION:	The function to be performed. Value must be the name of some other screen or one of following:
APPEND	Retrieves documentation data from the database and attaches documentation text to the end of any existing lines in the Active Area (text work area).
CLEAR	Clears screen input data fields and clears the user's Active Area.
DELETE	Deletes a documentation member from the database.
EDIT	Transfers the user to the edit facility and allows text processing. See Chapter 11, "Edit Facility" on page 11-1 and following subtopics for a discussion of the available commands for processing the desired text.
FE	Combination FETCH and EDIT.
FETCH	Retrieves documentation data from the database and replaces the user's Active Area with documentation text.
LIST	Lists documentation data only. In batch, a formatted screen is not listed; only a found or not found message is returned.
REPL	Replaces a documentation member in the database.
SAVE	Adds a new documentation member to the database.
UPD	Updates formatted screen fields only. It does not affect the documentation text.
	Required: Yes
	Batch keyword: Positional parameter
n/a	This field has a constant value of NWK for network documentation.
	Required: Yes for batch only
	Batch keyword: Positional parameter
JOB:	Indicates the job related to the network for the documentation being defined.
	Size/Type: 1 to 8 alphanumeric characters
	Required: No
	Batch keyword: JOB
SYSTEM:	Identifies the system name or system ID related to the network for the documentation being defined.
	Size/Type: 1 to 8 alphanumeric characters
	Required: No
	Batch keyword: SYSTEM

NETWORK:	<p>The network for which the documentation is being defined.</p> <p>Size/Type: 1 to 8 alphanumeric characters</p> <p>Required: Yes</p> <p>Batch keyword: NETWORK</p>
DESC:	<p>A description of the network.</p> <p>Size/Type: 1 to 45 alphanumeric characters</p> <p>Required: No</p> <p>Batch keyword: DESC</p>
LINK:	<p>The number of another documentation member to link to the documentation member listed. Linking documentation members together causes them to be listed together as if they were a single member. The LINK member is listed after the requested member when using the LPROS command only.</p> <p>Size/Type: 1 to 6 numeric characters</p> <p>Required: No</p> <p>Batch keyword: LINK</p> <p>Note: It is possible to create an LPROS loop. Linked documentation should not be linked to itself. The following example shows a loop:</p> <pre> A linked to B B linked to C C linked to A </pre>
The following fields are informational only:	
STEP:	<p>The name of a job step in a particular job for which the network documentation is being defined.</p> <p>Size/Type: 1 to 8 alphanumeric characters</p> <p>Required: No</p> <p>Batch keyword: STEP</p>
DDNAME:	<p>A ddname for which the documentation is being defined.</p> <p>Size/Type: 1 to 8 alphanumeric characters</p> <p>Required: No</p> <p>Batch keyword: DD</p>
REPORT-ID:	<p>Identification of a printed report.</p> <p>Size/Type: 1 to 20 alphanumeric characters</p> <p>Required: No</p> <p>Batch keyword: REPORT</p>
FORM:	<p>Output form number.</p> <p>Size/Type: 1 to 8 alphanumeric characters</p> <p>Required: No</p> <p>Batch keyword: FORM</p>

TRAIN:	Type of printer train required. Size/Type: 1 to 2 alphanumeric characters Required: No Batch keyword: TRAIN
CARRIAGE:	Type of carriage control required. Size/Type: 1 to 4 alphanumeric characters Required: No Batch keyword: CARR
COPIES:	Number of original copies required. Size/Type: 1 to 3 numeric characters from 0 to 255 Required: No Batch keyword: COPIES
ACTIVE SIZE:	A system-generated field which shows the number of lines of text existing in the Active Area for the current terminal session.

7.1.4 DB.4.3 - User-Defined Item Documentation Screen

Use this screen for defining user level documentation.

```

----- CA-7 USER-DEFINED ITEM DOCUMENTATION -----
FUNCTION: xxxxxxxx (APPEND,CLEAR,DELETE,EDIT,FE,FETCH,LIST,REPL,SAVE,UPD)

USER: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

DESC: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx LINK: nnnnnn

REPORT-ID: xxxxxxxxxxxxxxxxxxxxxxxx
FORM: xxxxxxxx TRAIN: xx CARRIAGE: xxxx COPIES: nnn

ACTIVE SIZE: nnnn

PROGRAM: SM11 MSG-INDX: nn -- DB.4.3 -- yy.ddd / hh:mm:ss
MESSAGE:

```

To display, enter:

- 3 as the FUNCTION on the DB.4 Menu screen.
- 4.3 as the FUNCTION on the DB Menu screen.
- DB.4.3 as the FUNCTION on any other menu or formatted input screen.
- DB.4.3 or PROSE,USER as a top line command.

To exit:

- Press the PF3 key to return to the DB.4 Menu screen except with the TSO/ISPF interface when PF3 has been assigned to some other function.
- Move the cursor to the top line of the screen and enter the desired top line command if yet another function is desired.

7.1.4.1 Field Descriptions

FUNCTION:	The function to be performed. Value must be the name of some other screen or one of the following:
APPEND	Retrieves documentation data from the data base and attaches documentation text to the end of any existing lines in the Active Area (text work area).
CLEAR	Clears screen input data fields and clears the user's Active Area.
DELETE	Deletes a documentation member from the database.
EDIT	Transfers the user to the EDIT facility and allows text processing. See Chapter 11, "Edit Facility" on page 11-1 and following subtopics for a discussion of the available commands for processing the desired text.
FE	Combination FETCH and EDIT.
FETCH	Retrieves documentation data from the database and replaces the user's Active Area with documentation text.
LIST	Lists documentation data only. In batch, a formatted screen is not listed; only a found or not found message is returned.
REPL	Replaces a documentation member in the database.
SAVE	Adds a new documentation member to the database.
UPD	Updates formatted screen fields only. It does not affect the documentation text.
	Required: Yes
	Batch keyword: Positional parameter
n/a	This field has a constant value of USER for user documentation.
	Required: Yes for batch only
	Batch keyword: Positional parameter

USER: A required name used to supply a unique identifier for the information being defined by the user documentation entry. This field must follow OS data set naming conventions and must not correspond to any system name, job name, data set name, job-step-ddname combination, or workstation name which exists (or will exist) in the CA-7 system.

Size/Type: 1 to 32 alphanumeric characters
 Required: Yes
 Batch keyword: USER

DESC: An optional description of the user entry.

Size/Type: 1 to 45 alphanumeric characters
 Required: No
 Batch keyword: DESC

LINK: The number of another documentation member to link to the documentation member listed. Linking documentation members together causes them to be listed together as if they were a single member. The LINK member is listed after the requested member when using the LPROS command only.

Size/Type: 1 to 6 numeric characters
 Required: No
 Batch keyword: LINK

Note: It is possible to create an LPROS loop. Linked documentation should not be linked to itself. The following example shows a loop:

```
A linked to B
B linked to C
C linked to A
```

The following fields are informational only:

REPORT-ID: Identification of a printed report.

Size/Type: 1 to 20 alphanumeric characters
 Required: No
 Batch keyword: REPORT

FORM: Output form number.

Size/Type: 1 to 8 alphanumeric characters
 Required: No
 Batch keyword: FORM

TRAIN: Type of printer train required.

Size/Type: 1 to 2 alphanumeric characters
 Required: No
 Batch keyword: TRAIN

CARRIAGE: Type of carriage control required.
Size/Type: 1 to 4 alphanumeric characters
Required: No
Batch keyword: CARR

COPIES: Number of original copies required.
Size/Type: 1 to 3 numeric characters from 0 to 255
Required: No
Batch keyword: COPIES

ACTIVE SIZE: A system-generated field which shows the number of lines of text existing in the Active Area for the current terminal session.

7.1.5 DB.4.4 - Data Set Documentation Screen

Use this screen for defining data set level documentation. Data sets whose name is longer than 41 characters cannot use this screen.

```

----- CA-7 DATA SET DOCUMENTATION -----
FUNCTION: xxxxxxxx (APPEND,CLEAR,DELETE,EDIT,FE,FETCH,LIST,REPL,SAVE,UPD)

DSN: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx DSNBR: nnnnnn

DESC: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx LINK: nnnnnn

STEP: xxxxxxxx DDNAME: xxxxxxxx
REPORT-ID: xxxxxxxxxxxxxxxxxxxxxxxx
FORM: xxxxxxxx TRAIN: xx CARRIAGE: xxxx COPIES: nnn

ACTIVE SIZE: nnnn

PROGRAM: SM11 MSG-INDX: nn -- DB.4.4 -- yy.ddd / hh:mm:ss
MESSAGE:

```

To display, enter:

- 4 as the FUNCTION on the DB.4 Menu screen.
- 4.4 as the FUNCTION on the DB Menu screen.
- DB.4.4 as the FUNCTION on any other menu or formatted input screen.
- DB.4.4 or PROSE,DSN as a top line command.

To exit:

- Press the PF3 key to return to the DB.4 Menu screen except with the TSO/ISPF interface when PF3 has been assigned to some other function.
- Move the cursor to the top line of the screen and enter the desired top line command if yet another function is desired.

7.1.5.1 Field Descriptions

FUNCTION:	The function to be performed. Value must be the name of some other screen or one of the following:
APPEND	Retrieves documentation data from the database and attaches documentation text to the end of any existing lines in the Active Area (text work area).
CLEAR	Clears screen input data fields and clears the user's Active Area.
DELETE	Deletes a documentation member from the database.
EDIT	Transfers the user to the EDIT facility and allows text processing. See Chapter 11, "Edit Facility" on page 11-1 and following subtopics for a discussion of the available commands for processing the desired text.
FE	Combination FETCH and EDIT.
FETCH	Retrieves documentation data from the database and replaces the user's Active Area with documentation text.
LIST	Lists documentation data only. In batch, a formatted screen is not listed; only a found or not found message is returned.
REPL	Replaces a documentation member in the database.
SAVE	Adds a new documentation member to the database.
UPD	Updates formatted screen fields only. It does not affect the documentation text.
	Required: Yes
	Batch keyword: Positional parameter
n/a	This field has a constant value of DSN for data set documentation.
	Required: Yes for batch only
	Batch keyword: Positional parameter
DSN:	The data set name for which the documentation is being defined.
	Size/Type: 1 to 41 alphanumeric characters
	Required: Yes (unless DSNBR is used)
	Batch keyword: DSN
	Note: The data set name must not exceed 41 characters to allow PP. to be added as part of the data set name in the IDs.

DSNBR: Identifies the data set for which the documentation is being supplied. Must be the data set number assigned by CA-7 when the data set was first defined.

Size/Type: 1 to 6 numeric characters
 Required: Yes (unless DSN is used)
 Batch keyword: DSNBR

DESC: An optional description of the data set.

Size/Type: 1 to 45 alphanumeric characters
 Required: No
 Batch keyword: DESC

LINK: The number of another documentation member to link to the documentation member listed. Linking documentation members together causes them to be listed together as if they were a single member. The LINK member is listed after the requested member when using the LPROS command only.

Size/Type: 1 to 6 numeric characters
 Required: No
 Batch keyword: LINK

Note: It is possible to create an LPROS loop. Linked documentation should not be linked to itself. The following example shows a loop:

A linked to B
 B linked to C
 C linked to A

The following fields are informational only:

REPORT-ID: Identification of a printed report.

Size/Type: 1 to 20 alphanumeric characters
 Required: No
 Batch keyword: REPORT

FORM: Output form number.

Size/Type: 1 to 8 alphanumeric characters
 Required: No
 Batch keyword: FORM

TRAIN: Type of printer train required.

Size/Type: 1 to 2 alphanumeric characters
 Required: No
 Batch keyword: TRAIN

CARRIAGE: Type of carriage control required.
Size/Type: 1 to 4 alphanumeric characters
Required: No
Batch keyword: CARR

COPIES: Number of original copies required.
Size/Type: 1 to 3 numeric characters from 0 to 255
Required: No
Batch keyword: COPIES

ACTIVE SIZE: A system-generated field which shows the number of lines of text existing in the Active Area for the current terminal session.

7.1.6 DB.4.5 - DD Statement Documentation Screen

Use this screen for defining DD level documentation.

```

- - - - - CA-7 DD STATEMENT DOCUMENTATION - - - - -
FUNCTION: xxxxxxxx  (APPEND,CLEAR,DELETE,EDIT,FE,FETCH,LIST,REPL,SAVE,UPD)


JOB: xxxxxxxx
SYSTEM: xxxxxxxx


DESC: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx LINK: nnnnnn


STEP: xxxxxxxx DDNAME: xxxxxxxx
REPORT-ID: xxxxxxxxxxxxxxxxxxxxx
FORM: xxxxxxxx TRAIN: xx CARRIAGE: xxxx COPIES: nnn


ACTIVE SIZE: nnnn


PROGRAM: SM11 MSG-INDX: nn -- DB.4.5 -- yy.ddd / hh:mm:ss
MESSAGE:
```

To display, enter:

- 5 as the FUNCTION on the DB.4 Menu screen.
- 4.5 as the FUNCTION on the DB Menu screen.
- DB.4.5 as the FUNCTION on any other menu or formatted input screen.
- DB.4.5 or PROSE,DD as a top line command.

To exit:

- Press the PF3 key to return to the DB.4 Menu screen except with the TSO/ISPF interface when PF3 has been assigned to some other function.
- Move the cursor to the top line of the screen and enter the desired top line command if yet another function is desired.

7.1.6.1 Field Descriptions

FUNCTION:	The function to be performed. Value must be the name of some other screen or one of the following:
APPEND	Retrieves documentation data from the database and attaches documentation text to the end of any existing lines in the Active Area (text work area).
CLEAR	Clears screen input data fields and clears the user's Active Area.
DELETE	Deletes a documentation member from the database.
EDIT	Transfers the user to the EDIT facility and allows text processing. See Chapter 11, "Edit Facility" on page 11-1 and following subtopics for a discussion of the available commands for processing the desired text.
FE	Combination FETCH and EDIT.
FETCH	Retrieves documentation data from the database and replaces the user's Active Area with documentation text.
LIST	Lists documentation data only. In batch, a formatted screen is not listed; only a found or not found message is returned.
REPL	Replaces a documentation member in the database.
SAVE	Adds a new documentation member to the database.
UPD	Updates formatted screen fields only. It does not affect the documentation text.
	Required: Yes
	Batch keyword: Positional parameter
n/a	This field has a constant value of DD for the DD documentation.
	Required: Yes for batch only
	Batch keyword: Positional parameter
JOB:	Indicates the job containing the ddname for the documentation being defined.
	Size/Type: 1 to 8 alphanumeric characters
	Required: Yes
	Batch keyword: JOB

DESC:	Description of the data set. Size/Type: 1 to 45 alphanumeric characters Required: No Batch keyword: DESC
LINK:	The number of another documentation member to link to the documentation member listed. Linking documentation members together causes them to be listed together as if they were a single member. The LINK member is listed after the requested member when using the LPROS command only. Size/Type: 1 to 6 numeric characters Required: No Batch keyword: LINK Note: It is possible to create an LPROS loop. Linked documentation should not be linked to itself. The following example shows a loop: A linked to B B linked to C C linked to A
STEP:	This field indicates the step of the JOB for the DD documentation being defined. Size/Type: 1 to 8 alphanumeric characters Required: Yes Batch keyword: STEP
DDNAME:	This field indicates the ddname for which the documentation is being defined. Size/Type: 1 to 8 alphanumeric characters Required: Yes Batch keyword: DD

The following fields are informational only:

REPORT-ID:	Identification of a printed report. Size/Type: 1 to 20 alphanumeric characters Required: No Batch keyword: REPORT
FORM:	Output form number. Size/Type: 1 to 8 alphanumeric characters Required: No Batch keyword: FORM

TRAIN:	Type of printer train required. Size/Type: 1 to 2 alphanumeric characters Required: No Batch keyword: TRAIN
CARRIAGE:	Type of carriage control required. Size/Type: 1 to 4 alphanumeric characters Required: No Batch keyword: CARR
COPIES:	Number of original copies required. Size/Type: 1 to 3 numeric characters from 0 to 255 Required: No Batch keyword: COPIES
ACTIVE SIZE:	A system-generated field which shows the number of lines of text existing in the Active Area for the current terminal session.

7.1.7 DB.4.6 - Application System Documentation Screen

Use this screen for defining system level documentation.

```

----- CA-7 APPLICATION SYSTEM DOCUMENTATION -----
FUNCTION: xxxxxxxx (APPEND,CLEAR,DELETE,EDIT,FE,FETCH,LIST,REPL,SAVE,UPD)

SYSTEM: xxxxxxxx

DESC: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx LINK: nnnnnn

ACTIVE SIZE: nnnn

PROGRAM: SM11 MSG-INDX: nn -- DB.4.6 -- yy.ddd / hh:mm:ss
MESSAGE:

```

To display, enter:

- 6 as the FUNCTION on the DB.4 Menu screen.
- 4.6 as the FUNCTION on the DB Menu screen.
- DB.4.6 as the FUNCTION on any other menu or formatted input screen.
- DB.4.6 or PROSE,SYS as a top line command.

To exit:

- Press the PF3 key to return to the DB.4 Menu screen except with the TSO/ISPF interface when PF3 has been assigned to some other function.
- Move the cursor to the top line of the screen and enter the desired top line command if yet another function is desired.

7.1.7.1 Field Descriptions

FUNCTION:	The function to be performed. Value must be the name of some other screen or one of the following:
APPEND	Retrieves documentation data from the database and attaches documentation text to the end of any existing lines in the Active Area (text work area).
CLEAR	Clears screen input data fields and clears the user's Active Area.
DELETE	Deletes a documentation member from the database.
EDIT	Transfers the user to the EDIT facility and allows text processing. See Chapter 11, "Edit Facility" on page 11-1 and following subtopics for a discussion of the available commands for processing the desired text.
FE	Combination FETCH and EDIT.
FETCH	Retrieves documentation data from the database and replaces the user's Active Area with documentation text.
LIST	Lists documentation data only. In batch, a formatted screen is not listed; only a found or not found message is returned.
REPL	Replaces a documentation member in the database.
SAVE	Adds a new documentation member to the database.
UPD	Updates DESC and LINK fields only. It does not affect the documentation text.
	Required: Yes
	Batch keyword: Positional parameter
n/a	This field has a constant value of SYS for system documentation.
	Required: Yes for batch only
	Batch keyword: Positional parameter
SYSTEM:	Identifies the system name or system ID for which the documentation entry is being defined.
	Size/Type: 1 to 8 alphanumeric characters
	Required: Yes
	Batch keyword: SYSTEM

DESC: An optional description of the system.

Size/Type: 1 to 45 alphanumeric characters

Required: No

Batch keyword: DESC

LINK: The number of another documentation member to link to the documentation member listed. Linking documentation members together causes them to be listed together as if they were a single member. The LINK member is listed after the requested member when using the LPROS command only.

Size/Type: 1 to 6 numeric characters

Required: No

Batch keyword: LINK

Note: It is possible to create an LPROS loop. Linked documentation should not be linked to itself. The following example shows a loop:

A linked to B
B linked to C
C linked to A

ACTIVE SIZE: A system-generated field which shows how many lines of text exist in the Active Area for the current terminal session.

7.2 Special Documentation Features

There are six major categories of user-defined documentation. Each category has its own DB.4 screen with which the documentation may be entered and maintained in the database. All categories are optional and are provided for user convenience. Each documentation member falls within one of six categories:

- Job name
- System name
- Network name
- User level
- Data set name
- DD statement

7.2.1 User-Defined Documentation Segments

Within each of the major documentation member types, the user can subdivide documentation text into segments. Segments are convenient because they can be accessed individually by name, presenting documentation only on a specific item or set of items.

The user names and defines segments by the use of special pound sign (#) control statements. The # control statements begin and end each segment, forming a **bookend** notation.

7.2.1.1 Reserved Segment Names

The system reserves some names for special use. You should not use segment names which would conflict with the following:

```
#END
#QDESC
#RESTART
#station-name
```

Discussions of these special purpose names follow later in this chapter.

7.2.1.2 Rules and Guidelines for Documentation Segmentation

The following rules and guidelines help you make more efficient use of the segmentation tool:

- Adopt a naming convention for segments. This adds ease of retrieval. See 7.2.1.11, “Documenting Naming Conventions” on page 7-32.
- A documentation member can contain one or more segments.
- Each segment must have a name. Segment names must conform to the following rules:

- The name is a maximum of 8 characters.
- Commas, as part of the name, are not allowed.
- Alphabetics, numerics, and special characters are allowed.
- A segment name must be unique within the documentation member. A segment name should be different from the documentation member name of which the segment is a part.

7.2.1.3 Defining Segment Names

User-defined segments can be an orderly method of documentation. By establishing standards related to segment names, a powerful tool results. To define a segment name, select an 8-character name meaningful to your environment; precede the name with a # sign. Enter the definition starting in column 1. For example:

```
#segment-name
```

The following is a list of potential segment names which could fit into the naming conventions for segments:

#DATECARD	Format of the date cards used in this job.
#progname	Description of programs used in the job.
#stepname	Description of job steps in the job.
#DISASTER	Instructions for disaster recovery procedures.
#RECOVERY	Recovery considerations for I/O errors, abends, and so forth.
#FILING	Instructions for filing input or output.
#SETUP	I/O media requirements of the job.
#CONTACTS	Names and numbers of personnel who support the job.
#DEVICES	I/O device requirements of the job.
#CONFLICT	Possible processing conflicts with other jobs.
#VALIDATE	Special manual verification requirements of the job.
#AFTERHRS	Special procedures occurring after the close of the business day.

Other segment names may be helpful or more meaningful and easy to use in any particular data center. Within the constraints of the few reserved segment names, the user has total flexibility in naming documentation segments.

7.2.1.4 Format of Documentation Segments

The format of documentation segments is simple. The first statement must be a #segment-name statement. The #segment-name begins in column 1. Immediately following the #segment-name statement is the documentation associated with the name. The documentation is in free-form and control characters are not needed. However, avoid beginning a documentation line with a #, which the system interprets as a segmentation control character and tries to treat it as a start or an end of a segment or subsegment.

To terminate the documentation for this segment, a #END statement is required. The #END statement begins in column 1 and the format of this statement is as follows:

```
#END[,segment-name]
```

For every segment name defined, there must be a #END control statement. It must occur before a new segment name can be defined. That is, the #END statement must precede the next occurrence of a #segment-name statement.

The following shows an example of documentation segments. In this case, the user is documenting jobs, and thus, is entering job-level documentation.

```
job documentation
.
.    usual job documentation goes here
.
#DATECARD
.
.    date control statement information goes here
.
#END,DATECARD
#RECOVERY
.
.    production recovery documentation goes here
.
#END,RECOVERY
#CONTACTS
.
.    names, telephone numbers of responsible personnel go here
.
#END,CONTACTS
```

7.2.1.5 Displaying Documentation Segments

If standard naming conventions have been adopted by the installation, displaying the documentation contained in the segments is easy.

For example, all date card instructions for all jobs could be entered in the database, within each job's JOB documentation, with a standard segment name of DATECARD. Whenever any question arises regarding the date card for a particular job, the user can display the documentation with the LPROS command, giving only the name of that job and the segment name DATECARD as follows:

```
LPROS,JOB=jobname,SEG=DATECARD
```

If there is more documentation on this job, it is not displayed. Only the segment requested is displayed. To obtain all documentation on this job, omit the SEG parameter.

By segmenting documentation, the user can view selected parts of the documentation and not have to view all the documentation at one time.

7.2.1.6 Subsegments

Inside each segment may be subsegments. There are rules and guidelines which apply to subsegments and they are similar to those for segments.

7.2.1.7 Rules and Guidelines for Subsegments

- Adopt a naming convention for subsegments. Just as naming conventions aid the user in retrieval of segments, the same holds true with subsegments. See 7.2.1.11, “Documenting Naming Conventions” on page 7-32.
- Each subsegment must have a name. Subsegment names must follow the same rules as segment names. See 7.2.1.2, “Rules and Guidelines for Documentation Segmentation” on page 7-26.
- A segment may have one or more subsegments.
- A subsegment name must be different from the segment name of which the subsegment is a part.
- A subsegment must end before the end of the segment of which it is a part.
- A subsegment name should not be the same as a reserved segment name. See 7.2.1.1, “Reserved Segment Names” on page 7-26.
- A subsegment must be embedded between a #subsegment-name statement and a #END,subsegment-name statement.

7.2.1.8 Defining Subsegment Names

To define a subsegment name, select an 8-character name meaningful in your environment; precede the name with a # sign. Enter the definition starting in column 1. This is exactly the same way segment names are defined:

```
#subsegment-name
```

For example, your installation may have two procedures for recovery: one for business hours and one for nonbusiness hours. To define these two names, you might create two subsegments within the segment named RECOVERY using the following subsegment names:

```
#DAYTIME
#AFTERHRS
```

7.2.1.9 Format of Documentation Subsegments

The format of documentation subsegments is exactly like that of segments. The first statement of the subsegment must be the `#subsegment-name` and begin in column 1. Immediately following this statement is the subsegment documentation which is free-form. Control characters are not needed. Avoid beginning a documentation line with a `#` sign, which the system interprets as a segmentation control character and tries to treat it as a start or an end of a subsegment or segment.

To terminate the documentation for this subsegment, a `#END` statement is required. The `#END` statement begins in column 1 and the format of this statement is as follows:

```
#END,subsegment-name
```

For every subsegment name defined, there must be a `#END,subsegment-name` control statement. Nesting of subsegments within other subsegments is not allowed. The `#END,subsegment-name` statement must occur before the next occurrence of a `#subsegment-name` or `#segment-name` statement.

The following shows an example of coding DAYTIME and AFTERHRS subsegments within the RECOVERY segment:

```
#RECOVERY
#DAYTIME
.    Daytime procedures go here
.
#END,DAYTIME
#AFTERHRS
.    After hours procedures go here
.
#END,AFTERHRS
#END,RECOVERY
```


7.2.1.10 Displaying Documentation Subsegments

If standard naming conventions for subsegments have been adopted by the installation, displaying the documentation contained in the subsegments is easy. The user must know which names are segment names and which names are subsegment names. This is where standard naming conventions for both is important.

In the previous example, in the segment named RECOVERY there are two subsegments; one is named DAYTIME and the other AFTERHRS. To display the subsegments, use the LPROS command. For example:

```
LPROS,JOB=jobname,SEG=(RECOVERY,AFTERHRS)
```

This would display only the AFTERHRS subsegment. The SEG parameter requires that the segment name appear first, followed by the subsegment name.

To display the other subsegment of the RECOVERY segment, the entry would be as follows:

```
LPROS,JOB=jobname,SEG=(RECOVERY,DAYTIME)
```

The DAYTIME subsegment would display.

To display all subsegments of a segment, only the segment form of LPROS is used. For example:

```
LPROS,JOB=jobname,SEG=RECOVERY
```

7.2.1.11 Documenting Naming Conventions

As a further aid in handling documentation, we recommend that not only should you adopt a naming convention for segments and subsegments but also that you document the naming convention. We suggest that you use at least one documentation member (not a segment) of the user-designated level for this purpose. The following are suggestions for naming a user documentation member:

```
STANDARD.NAMING.SEGMENTS  
STANDARD.NAMES.FOR.SEGMENTS
```

(See the USER field on the 7.1.4, “DB.4.3 - User-Defined Item Documentation Screen” on page 7-11.)

If your installation is using subsegments and segments, two documentation members, one for segments and one for subsegments, might be advisable. Following are suggestions for the documentation member name containing documentation on subsegment naming conventions:

```
STANDARD.NAMING.SUB.SEGMENTS  
STANDARD.NAMES.FOR.SUB.SEGMENTS
```

As with any documentation, the contents of documentation member(s) can be maintained easily through the appropriate DB.4 screens.

7.2.2 Special Purpose Documentation

The Special Purpose Documentation (#) feature allows the user to define documentation information to be used for special situations. The user must use the reserved names provided for these Special Purpose Documentation items. Once the information has been entered, it is available through the LPROS command. Optionally, it may also be displayed by automatically routing the information to a specific workstation or terminal.

Special Documentation is entered as a segment within a stream of regular documentation lines using the appropriate DB.4 screen (and the EDIT facility).

7.2.2.1 General Format

The general format of the Special Purpose Documentation entry (or # entry) is as follows with the # sign always in position 1 of the segment delimiter statements.

```
#reserved special purpose name
.
.   special documentation lines
.
#END,reserved special purpose name
```

7.2.3 #QDESC Segment Usage

One feature of the Special Purpose Documentation facility applies to job and network documentation. The following paragraphs provide explanations and examples of these Special Documentation capabilities.

If the user wants special job or network documentation to be automatically routed to a predetermined terminal when the task enters the queue, the reserved documentation segment name #QDESC should be used, within the job or network documentation. An example follows with the # sign always in position 1 of the segment delimiter statements.

```
#QDESC
.
.   special documentation for the job or network
.
#END,QDESC
```

When the job enters the queue, any QDESC documentation for the task is automatically routed to the terminal defined on the DB.1 screen (LTERM value), or the initialization file JCL statement, LTERM value, or if neither of these is specified, to the MASTER station. When a network enters the preprocessing or postprocessing queue, any QDESC documentation is automatically routed to the MASTER station. Optionally, #QDESC lines may be listed by the LPROS transaction with job name or network specified.

Note: Module SASSMSGSGS can cause suppression of the #QDESC. Check messages SFEP-11 and SIRD-11 in the SASSMSGSGS module.

7.2.4 #station-name Segment Usage

The user may want special network documentation to be automatically routed to predetermined workstations when the network enters the queue. The desired station names are defined as part of the documentation with a #station-name statement. The station name which is to receive the documentation would be placed in the #station-name statement immediately following the #. In this case, a #station-name record, indicating the name of the station, should be entered immediately ahead of the #QDESC record, as follows:

```
(A) #station-name1
(B) #QDESC
    .
    special network documentation for station-name1
    .
    #END,station-name1
(A) #station-name2
(B) #QDESC
    .
    special network documentation for station-name2
    .
    #END,station-name2
(A) #station-namen
(B) #QDESC
    .
    special network documentation for station-namen
    .
    #END,station-namen
```

Documentation lines may be entered between points (A) and (B), but these lines would only be displayed through the LPROS command or DB.4 screens. The lines between #QDESC and #END, however, are automatically sent to the indicated workstations when the network enters the queue.

Note: If #QDESC documentation is to be displayed at a station that is not a printer, module SASSMSGSGS must be changed for message SFEP-11. LT=*PRNTR* must be changed to either LT=*ALL* or LT=station-name. Also notice that station names should match station names defined in the STANIDS keyword of the STATIONS statement in the initialization file.

7.2.5 #RESTART Segment Usage

If the user wants to document recovery or restart instructions for a job, the reserved segment name #RESTART should be used. The restart documentation should be included in the DB.4.1 documentation as follows:

```
.
job level documentation
.
.
#RESTART
.
.
special restart/recovery documentation
.
.
#END,RESTART
.
.
job level documentation
.
.
.
```

As with all special documentation, the # sign must be in position 1 of the record as shown.

The system recognizes #RESTART as a system-defined segment name having a special use with the LIST command. It can also be used with the LPROS command, but #RESTART is treated as any other segment name.

When using the LIST command, if PROSE=YES is specified, the #RESTART documentation segment of the job documentation is displayed. PROSE=NO is the default.

With the LPROS command, RESTART has to be specifically requested. For example:

```
LPROS,JOB=jobname,SEG=RESTART
```

7.2.6 #END Statement

This statement is used to signal the end of a documentation unit. The format is as follows:

```
#END[,segment-name]
      [,subsegment-name]
      [,station-name]
```

Where:

#END

Is required. It must begin in position 1 of the record. If used without any other parameter, it marks the end of the documentation unit (that is, current segment and subsegment).

segment-name | subsegment-name | station-name

Identifies a user-defined segment, subsegment, or station name. It must match the name used in the # record at the beginning of the unit.

Depending on how documentation is structured by the user, multiple #END statements may be nested within one segment of documentation. For additional reference, see 7.2.1, “User-Defined Documentation Segments” on page 7-26.

Chapter 8. Workstation Networks

This chapter describes setting up and maintaining a workstation network within CA-7.

8.1 Adding a Network

Adding a network is similar to adding a job. The primary method is to use the ADD function on the DB.5 screen. For new users or users making multiple changes to their networks, other methods can be used.

If the network information already exists in machine-readable form, then writing a program to produce batch terminal network commands may be an advantage. The database can then be updated using the Batch Terminal Interface (BTI) facility. For further details on BTI, see 1.5, “Batch Input” on page 1-22 and the *CA-7 Interfaces Guide*.

Some items to remember when adding a network are:

1. On the DB.5 screen, stations must be defined in the chronological order in which they are used.
2. The DMDNW and RUNNW commands do not add a network to the database. These commands require that the network be defined in the database with some type of schedule before the commands can act on the network.
3. Input networks can be connected to one or more CPU jobs, in which case, they have a predecessor/dependency relationship with those jobs. These connections are defined using the DB.3.4 screen functions.
4. Input networks which are not to be handled on an on-request basis must have a calendar-based schedule defined. This is done with the DB.2.2 screen.
5. Output networks are scheduled to be run because of their connection to a job. When the connected CPU job is scheduled into the queue, the output network is brought into the postprocess queue.
6. Output networks must be logically connected with the DB.3.4 screen. Scheduling parameters must be entered through the DB.2.3 screen, but the schedule is not resolved against a calendar.

8.2 DB.5 - Input/Output Network Definition Screen

Use this screen to define networks of workstations at which manual tasks are performed.

```

----- CA-7 INPUT/OUTPUT NETWORK DEFINITION -----
FUNCTION: xxxxxxxx (ADD,DELETE,FORMAT,LIST,UPD)

NETWORK: xxxxxxxx TYPE: xxxxxx
SUB-ID: xxxxxxxx JOB: xxxxxxxx SCHD PROSE: xxxxxxxx

STATION 1: xxxxxxxx
STATION 2: xxxxxxxx
STATION 3: xxxxxxxx
STATION 4: xxxxxxxx
STATION 5: xxxxxxxx
STATION 6: xxxxxxxx
STATION 7: xxxxxxxx
STATION 8: xxxxxxxx
STATION 9: xxxxxxxx

PROGRAM: SM40 MSG-INDX: nn -- DB.5 -- yy.ddd / hh:mm:ss
MESSAGE:

```

To display, enter:

- 5 as the FUNCTION on the DB Menu screen.
- DB.5 as the FUNCTION value on any other menu or formatted input screen.
- DB.5 or NETWORK as a top line command.

To exit:

- Press the PF3 key to return to the DB Menu screen except with the TSO-ISPF interface when PF3 has been assigned to some other function.
- Move the cursor to the top line of the screen and enter the desired top line command if yet another function is desired.

8.2.1.1 Field Descriptions

FUNCTION:	The function to be performed. Value must be the name of some other screen or one of the following:
ADD	Adds a network to the database.
DELETE	Deletes a network, and its associated documentation and schedules, from the database.
	Note: A network cannot be deleted if it is connected to a job.
FORMAT	Clears the screen of all user input data.
LIST	Lists the database information. In batch, a formatted screen is not listed; only a found or not found message is returned.
UPD	Updates network information in the database.
	Required: Yes
	Batch keyword: Positional parameter
NETWORK:	The network on which the indicated function is to be performed. Must be a network name.
	Size/Type: 1 to 8 alphanumeric characters
	Required: Yes
	Batch keyword: Positional parameter
TYPE:	The value must be INPUT or OUTPUT to indicate the type of network.
	Required: Yes - for ADD and UPD functions No - for DELETE, FORMAT, and LIST functions
	Batch keyword: Positional parameter
SUB-ID:	Used for further description of the network's duties.
	Size/Type: 1 to 8 alphanumeric characters
	Required: No
	Batch keyword: SUBID
	Output Networks: SUBID should be specified on the DB.3.4 connection. Then, this sub-ID is shown on subsequent inquiries.
	Input Networks: The SUBID specified here shows on various inquiries. For example, if handling of input data is required for the network, the user's task number, job number, or control number for the data may be defined as the sub-ID.

JOB: A job name. For this job, the schedule associated with the output network is assumed to be accurate and is not adjusted based on job due-out time. For all other connected jobs, the network's scheduling parameters for time-of-day are adjusted as if the network entered the postprocessing queue at the job's due-out time.

Size/Type: 1 to 8 alphanumeric characters
 Required: No
 Batch keyword: JOB

SCHD PROSE: This name designates the name of some other network whose DB.4.2 member, #QDESC segment, is to be used for special documentation messages which are printed by Schedule Scan when this network enters the queue. Cannot be used if this network has a DB.4.2 member defined for it.

Size/Type: 1 to 8 alphanumeric characters
 Required: No
 Batch keyword: SCHPROSE

STATION: These fields allow entry of up to nine workstations into this network. Station names must be entered in the sequence in which the corresponding workstation tasks are to be performed. The same station name may be used more than once, but the names must have a correlation to the STANIDS or LTERM names as defined in the initialization file. These station names **must not** conflict with any existing network names. For BTI input, the station names should be a sublist. For example:

STAT=(st1,st2,st3)

Size/Type: 1 to 8 alphanumeric characters
 Required: At least one entry
 Batch keyword: STAT

Note: To remove a station, place an asterisk (*) in that STATION field and use the UPD function. However, if the network has a schedule, the number of stations in a network cannot be changed by an update.

8.2.1.2 Batch Input Example

This batch input example adds input network PAYPUNCH to the CA-7 database. The network consists of three stations: LOG, KEY, and VERIFY.

```
NETWORK
ADD,PAYPUNCH,INPUT,STAT=(LOG,KEY,VERIFY)
DBM
```

8.3 Changing a Network

You should use the UPD function on the DB.5 screen if the characteristics of the network are changing from that shown on the DB.5 screen. If there are many changes, you can make these changes by using the Batch Terminal Interface (BTI) facility. For further details on BTI, see 1.5, “Batch Input” on page 1-22 and the *CA-7 Interfaces Guide*. In most cases, reconnecting the network may not be necessary, but the schedule may be different. Be sure to consider the schedule function for this particular network. You can use the DB.2 scheduling screens for this purpose.

If the function of the network changes, the job connections for that network may also change. In this case, you can use the DB.3.4 or DB.3.5 screen to redefine the job connections.

Keep in mind that one network can be connected to more than one CPU job and schedule changes may be associated with the change of network function. Use the `LNTWK,NW=networkname,LIST=USERS` command to determine which jobs are connected to the network.

If the number of stations in a network changes, the stations cannot be added or deleted if the network has a schedule. The schedule must first be deleted.

Chapter 9. Data Sets

This chapter contains instructions for adding and maintaining data set information in CA-7.

9.1 Adding a Data Set

There are several methods of adding data sets to the database. The primary method is accomplished through the LOAD process. When a job is loaded, all data set names referenced in the JCL are added to the database if those names are not already in the database. If the LOADDSENS keyword is used on the DBASE statement in the initialization file, then the LOAD process does not build any DD or data set information for jobs that are marked MAINT=Y on the DB.1 (JOB) screen.

Data sets can also be added to the database by using the DB.6 screen ADD function.

If data set information already exists in machine-readable form, a program can be written to perform the DB.6 screen functions with batch commands and the Batch Terminal Interface (BTI) facility. For further details on BTI, see 1.5, “Batch Input” on page 1-22 and *CA-7 Interfaces Guide*.

Once a data set is added to the database, CA-7 assigns a unique 6-digit number to the data set, referred to as a DSNBR. The data set can then be referenced by that number. This eliminates the need to enter a long data set name each time the data set is to be referenced by CA-7 commands and screens.

9.1.1 Permanent Data Sets

During the LOAD process, any data set that is used as input to the job is considered a requirement for that job. This means that a new version of the data set must be created before that job can be submitted by CA-7. Some data sets, however, are not created by a CA-7 submitted job. The user may wish these data sets to be considered as permanent to CA-7.

The LOAD process automatically adds data sets to the database and marks them permanent when all of the following conditions are true:

- Data set name does not already exist in the database.
- Data set is normal data set (that is, not temporary, SYSIN, or SYSOUT).
- Data set is accessed using a reserved ddname:

STEPLIB
SORTLIB
JOBLIB
STEPCAT
JOBCAT
SYSUDUMP
SYSABEND
SYSMDUMP
SYSCHK
SYSCKEOV

Reserved ddnames reside in the SASSPMDD table. Names can be added or deleted as necessary to satisfy the needs of the user. See the topic Reserved DDNAME Table in the "User Exits and Modifications" chapter of the *CA-7 Systems Programmer Guide* for a discussion of modifying this table.

The TYPE field on the DB.6 screen must be set to PERM for permanent data sets to CA-7. This must be done manually in those cases where the LOAD process could not determine that the data set was permanent. When the data set is defined as permanent, it is not considered to be a requirement for any job that uses it as input.

9.1.2 Dynamically Allocated Data Sets

Data sets which are dynamically allocated by the program(s) and do not have a DD statement in the JCL must be defined with the DB.6 screen. The LOAD process cannot perform this function since a DD statement was not used.

9.1.3 External Data Sets

Data sets can be defined as preexecution requirements for jobs even though the data set is never created or referenced by any job running under CA-7's control. Such data sets are referred to as external data sets and must be defined using the DB.6 screen. Data sets which are used by jobs under CA-7's control but are never created with a DISP=NEW are also referred to as external data sets. These data sets can be defined as job requirements using the DB.3.1 screen, but the requirements must be satisfied by the user before the dependent job can execute. Requirement posting can be performed using online screens, batch terminal interface commands, Trailer Step commands, or U7SVC input.

9.2 DB.6 - Data Set Definition Screen

Use this screen to define a user data set to CA-7.

Normally, you use this screen to define data sets with a TYPE value of PERM or for updates. If this screen is not used to define a data set, the initial LOAD of the job adds the data set to the database automatically. You can also perform any manual maintenance required on this information with this screen.

```

----- CA-7 DATA SET DEFINITION -----
FUNCTION: xxxxxxxx (ADD,DELETE,FORMAT,LIST,RENAME,UPD)

DSN: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx DSNBR: nnnnnn

NEWNAME: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

TYPE: xxxx    GDG: x

SMF FEEDBACK REQUIRED: x          POST AT CLOSE TIME: x

DEVICE: xxxx
DSORG: xxx
RECFM: xxxxx
LRECL: nnnnn
BLKSIZE: nnnnn

PROGRAM: SM30 MSG-INDX: nn  -- DB.6    -- yy.ddd / hh:mm:ss
MESSAGE:

```

To display, enter:

- 6 as the FUNCTION on the DB Menu screen.
- DB.6 as the FUNCTION value on any other menu or formatted input screen.
- DB.6 or DSN as a top line command.

To exit:

- Press the PF3 key to return to the DB Menu screen except with the TSO-ISPF interface when PF3 has been assigned to some other function.
- Move the cursor to the top line of the screen and enter the desired top line command if yet another function is desired.

9.2.1.1 Field Descriptions

FUNCTION:	The function to be performed. Value must be the name of some other screen or one of the following:
ADD	Adds a data set to the database.
DELETE	Deletes a data set and its associated elements (data set triggers, user documentation, and so forth).
	Note: If any job uses this data set, it cannot be deleted until the jobs or their DSN predecessor requirements are deleted.
FORMAT	Clears the screen of all user input data.
LIST	Lists the formatted data of the screen. In batch, a formatted screen is not listed; only a found or not found message is returned.
RENAME	When the user has changed a data set's name, RENAME with the NEWNAME field may be used to change the CA-7 data set name. The RENAME top line utility command changes the data set name in OS; whereas, the RENAME function on this screen changes the data set name in the CA-7 database, but not in OS.
UPD	Updates data set information in the database.
	Required: Yes
	Batch keyword: Positional parameter
DSN:	Identifies the data set and must be the fully qualified data set name.
	Size/Type: 1 to 44 alphanumeric characters
	Required: Yes - for ADD (and UPD unless DSNBR is used)
	Batch keyword: DSN
	Note: CA-7 does not support DSNs with embedded blanks in the name, or any other nonconventional data set name.
DSNBR:	Identifies the data set to be used and must be the numeric value already assigned by CA-7.
	Size/Type: 1 to 6 numeric characters (leading zeros not required)
	Required: Yes - unless DSN is used (ignored for ADD)
	Batch keyword: DSNBR

NEWNAME: Used with the RENAME function and DSN field to change a data set name in the CA-7 database. The RENAME top line utility command changes the data set name in OS; whereas, the RENAME function on this screen changes the data set name in the CA-7 database, but not in OS.

Size/Type: 1 to 44 alphanumeric characters
 Required: No
 Batch keyword: NEWNAME

TYPE: Indicates the data set type. Value can be:

NORM Internal: Both the creating-job and using-jobs for this data set are known to CA-7.

External: The creating-job and/or using-job(s) for this data set is not known to CA-7.

PERM This data set is always available for input. See 9.1.1, “Permanent Data Sets” on page 9-3.

Note: When a data set is marked PERM, no SMF records are captured, so no data set triggering can occur with this type data set.

Default: NORM when doing ADD
 Required: No
 Batch keyword: TYPE

GDG: Indicates whether this data set is a generation data group (Y or N).

Default: N
 Required: No
 Batch keyword: GDG

Note: Specific GDG creations cannot be used for posting requirements or for triggering jobs. Posting and triggering are done based on the creation/updating (SMF 15 record) of any generation of the GDG.

SMF FEEDBACK REQUIRED:

Indicates whether the interface to System Management Facility (SMF) should insert the values for DSORG, RECFM, LRECL and BLKSIZE when this data set is next used (Y or N). Unless reset, this insertion is only done once.

Default: Y
 Required: No
 Batch keyword: SMF

Note: After the first time the job that creates this data set runs under CA-7, this field is reset to N. Should the data set attributes change, this field must be reset to Y and the attribute field that changed (that is, DSORG, RECFM) must be zeroed (if numeric) or blanked out. This causes CA-7 to record the new values when the job is run again.

POST AT CLOSE TIME:

Indicates when to post the creation or update of a data set to the database and queues (Y or N). If Y is specified, action takes place as soon as the SMF data set creation record is received (when the data set is closed). This does not guarantee successful step or job termination.

Size/Type: Y or N
 Default: N (causes triggering or posting action to take place at job termination)
 Required: No
 Batch keyword: POST

DEVICE:

Indicates device type. If specified, TAPE or DASD are the only acceptable values.

Required: No
 Batch keyword: DEVICE

DSORG:

Indicates the data set organization. Same as the DSORG specified in the DCB subparameter in the JCL.

Size/Type: 1 to 3 alphanumeric characters
 Required: No
 Batch keyword: DSORG

RECFM:

Indicates the record format of the data set. Same as the RECFM specified in the DCB subparameter in the JCL.

Size/Type: 1 to 5 alphanumeric characters
 Required: No
 Batch keyword: RECFM

LRECL:

Indicates the logical record length of the data set. Same as the LRECL specified in the DCB subparameter in the JCL.

Size/Type: 1 to 5 numeric characters
 Required: No
 Batch keyword: LRECL

BLKSIZE:

Indicates the block size of the data set. Same as the BLKSIZE specified in the DCB subparameter in the JCL.

Size/Type: 1 to 5 numeric characters
 Required: No
 Batch keyword: BLKSIZE

9.2.1.2 Usage Notes

You can maintain connections to jobs that require this data set using the DB.3.1 screen.

9.2.1.3 Batch Input Example

This batch input example adds data set SYS2.PROCLIB to the CA-7 database as a permanent type.

```
DBM
DSN
ADD,DSN=SYS2.PROCLIB,TYPE=PERM
DBM
```

9.3 Changing a Data Set

The DB.6 screen is used to change the characteristics of a data set by entering the UPD function and the changed field values.

One of the more common uses of this screen is that of renaming data sets. If the DCB for this data set is not changing, all that needs to be entered is the RENAME function and the NEWNAME for the data set. This changes the data set name in the CA-7 database. To change the data set name in OS, use the RENAME top line utility command.

If DCB attributes change, be sure that the SMF indicator has the proper value for the situation. If the system is to automatically pick up the DCB characteristics, the SMF indicator should be set to Y and the appropriate DCB field cleared to blanks or zeros.

Note: RENAME should occur before changing JCL and LOADING of jobs; otherwise, DB.3.1 updates may need to be manually performed to disconnect the old data set name.

Chapter 10. JCL Management

This chapter identifies the screens and commands used to manage JCL within CA-7. It also includes discussion of JCL validation, overrides, LOAD command processing, the SASSJCLU utility program, and use of alternate libraries.

10.1 DB.7 - JCL Library Maintenance Screen

Use this screen to enter or review JCL statements in PDS or sequential JCL libraries. It can also be used to review CA-Librarian or CA-Panvalet JCL.

```

----- CA-7 JCL LIBRARY MAINTENANCE -----
FUNCTION  xxxxxxxx  (APPEND,CLEAR,DELETE,EDIT,FE,FETCH,
                     RENAME,REPL,RUN,RUNH,SAVE)

MEMBER  xxxxxxxx  NEWNAME/OPTION: xxxxxxxx

DSN  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx  JCL-ID: nnn
VOLSER  xxxxxxxx  JCLLIB: &xxxxxxxxxxxxxxxxx

ACTIVE SIZE  nnnn


PROGRAM  SM50  MSG-INDX: nn  -- DB.7  -- yy.ddd / hh:mm:ss
MESSAGE

```

To display, enter

- 7 as the FUNCTION on the DB Menu screen.
- DB.7 as the FUNCTION on any other menu or formatted input screen.
- DB.7 or JCL as a top line command.

To exit

- Press the PF3 key to return to the DB Menu screen except with the TSO/ISPF interface when PF3 has been assigned to some other function.
- Move the cursor to the top line of the screen and enter the desired top line command if yet another function is desired.

10.1.1 Field Descriptions

FUNCTION: The function to be performed. Value must be the name of some other screen or one of the following:

APPEND Attaches card-image data to the end of any existing lines in the terminal's Active Area.

CLEAR Clears screen input data fields and clears the user's Active Area.

DELETE Deletes a member from a PDS.

EDIT transfers the user to the edit facility and allows text processing. See Chapter 11, "Edit Facility" on page 11-1 and following subtopics for a discussion of the available commands for processing the desired text.

FE Combination of FETCH and EDIT.

FETCH Retrieves data and replaces the user's Active Area with this text.

RENAME Used with the NEWNAME field to rename a PDS member.

REPL Replaces a PDS member or sequential data set with the contents of the Active Area.

RUN Submits the text editor Active Area to the default MAINID without verifying the input requirements or performing the updates which normally follow successful job completion. Job start and job end messages are received at the issuing terminal where RUN was entered. Jobs scheduled with this function always show a due-out date and time of 69001/0000. If a /*PRIORITY statement is found as the first statement being submitted by this function, the statement is ignored and not submitted.

Note: The SASSXX05 exit is invoked for each JCL statement. Also, if there is a job in the CA-7 database with this member name and it is set to insert the CA-11 RMS step and the CA-11 interface is installed, the RMS step is inserted.

RUNH Same action as RUN function except this job is placed in CA-7 hold status.

SAVE Adds a new PDS member or replaces a sequential data set with the contents of the Active Area.

Note: The SAVE or REPL function cannot be performed on CA-Librarian or CA-Panvalet files.

Required: Yes
Batch keyword: Positional parameter

MEMBER: For PDS, CA-Panvalet, or CA-Librarian access, the required member name. Must be omitted for sequential files.

Size/Type: 1 to 8 alphanumeric characters
Required: Yes
Batch keyword: Positional parameter

NEWNAME/OPTION:

Multiple purpose field. Required with the RENAME function to indicate the new name which is to replace the old name.

Size/Type: 1 to 8 alphanumeric characters
Default: Blank
Required: Yes for RENAME function only
Batch keyword: NEWNAME

Note: For CA-Librarian or CA-Panvalet data set functions, a value of N indicates that include statements are not to be expanded. Default causes those statements to be expanded.

DSN: The optional data set name. When reviewing a member of a PDS, CA-Panvalet, or CA-Librarian file, DSN is not needed if JCLID is specified. If the data set is cataloged, DSN is sufficient. If the data set is not cataloged, DSN must be specified with the VOLSER field below.

Size/Type: 1 to 44 alphanumeric characters
Required: No
Batch keyword: DSN

JCL-ID: JCL library identification. Can be used instead of DSN and VOLSER fields to refer to a JCL statement. Must be a numeric INDEX assigned to a JCL statement. JCL-ID and JCLLIB are mutually exclusive.

Size/Type: 1 to 3 numeric characters from 0 to 254. A value of 255 is reserved for symbolic indexes. See JCLLIB.
Default: Value is obtained from the corresponding job entry in the database if it exists. Otherwise, the default is 0 (if DSN and VOLSER are omitted).
Required: No
Batch keyword: JCLID

JCLLIB	JCL library identification. Can be used instead of DSN and VOLSER fields to refer to a JCL statement. Must be a symbolic INDEX assigned to a JCL statement. JCLLIB and JCL-ID are mutually exclusive.
Size/Type:	2 to 16 alphanumeric characters beginning with an ampersand (&). The value &HELP is reserved.
Default:	Blank
Required:	No
Batch keyword:	JCLLIB
VOLSER:	Indicates the volume serial number on which the data set resides. Only used with DSN above. The volume must be available to CA-7.
Size/Type:	1 to 6 alphanumeric characters
Required:	No
Batch keyword:	VOL
ACTIVE SIZE:	A system-generated field which tells the user how many lines exist in the Active Area for the current terminal session.
Note:	CA-Panvalet, CA-Librarian, and PDS data sets require a JCL definition in the initialization file and in the execution JCL for CA-7.
	When EDIT or FE is entered, the user is transferred to the edit facility. See Chapter 11, "Edit Facility" on page 11-1 for a complete description of this feature.

10.1.2 Usage Notes

A shortcut method of displaying a specific job is to enter JCL,jobname as a top line command.

You can use this screen to access any card-image PDS, CA-Librarian, CA-Panvalet, or sequential file which is on a DASD volume available to CA-7.

If a SAVE/REPL is done for a member name that matches a job name in the CA-7 database, and the specified DSN library is a CA-7 JCL data set, then the job is flagged for reload on the next run. However, if RELOAD of N is used on the DB.1 (JOB) screen, then the RELOAD on the next run is ignored.

10.2 Scheduled Overrides

CA-7 provides the ability to schedule JCL overrides. Scheduled overrides may consist of any statement that may be included with the JCL for a job, not just JCL statements. Some special purpose statements, described later in this chapter, and any JCL or embedded data records can be scheduled. This enables users to have the run stream contents dynamically constructed to fit the needs of any particular run.

Overrides may be scheduled to be used:

- on specific dates
- after specific dates
- before specific dates
- between specific dates
- only after a certain time-of-day
- only before a certain time-of-day
- with specific schedule IDs
- within a range of schedule IDs
- only when CA-7/RPT is up or down

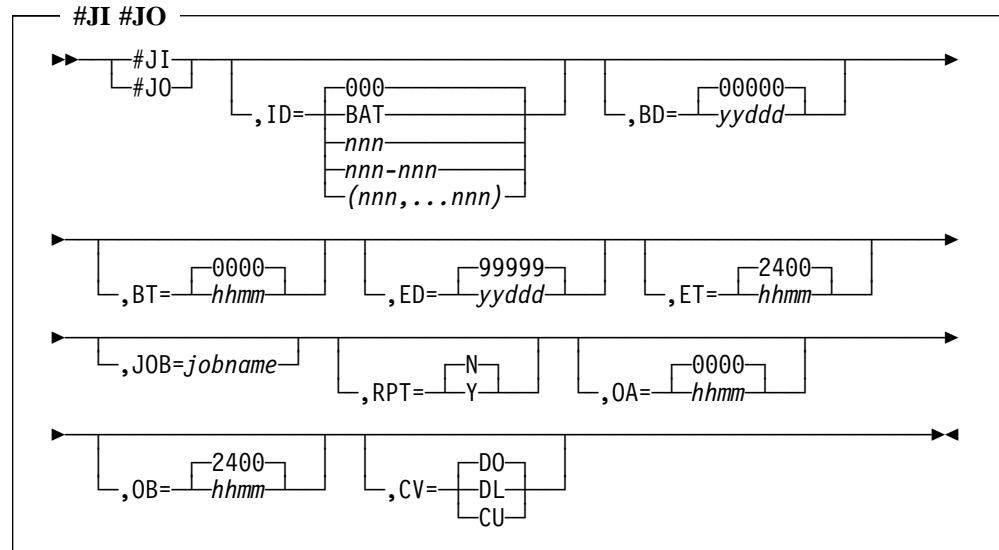
This enables users to prepare the proper statements anytime in advance of their needs without having to remember them on the day(s) they are needed. This also enables users to ensure that they are used only on the appropriate day(s).

Scheduled overrides are placed in the execution JCL members. They cannot be used in PROCLIB members since CA-7 does not access those members directly. The statements to be scheduled are placed in the JCL wherever they belong and are bookended with special reserved CA-7 control statements. One special statement is placed immediately ahead of the statements to define the scheduling criteria and another special statement is placed at the end of the set to mark the end of the statements being scheduled. Multiple sets may be included within a single job. All sets must be located after the JOB statement.

The first special statement of each set indicates the scheduling criteria and whether the other statements in the set are to be included or omitted based on those criteria. These special statements are identified by the characters #JI, #JO, #XI, or #XO in positions 1-3 of the statement. Also, these statements cannot be continued.

10.2.1 #JI and #JO Statements

10.2.1.1 Syntax



Where:

#JI

Indicates the statements are to be included based on the specified scheduling criteria.

#JO

Indicates the statements are to be omitted based on the specified scheduling criteria.

ID

Indicates a schedule ID number. If the job is scheduled by this ID, this test for statement inclusion or exclusion is considered true.

Default: 000

Required: No

000

All schedule IDs.

nnn

Indicates a specific schedule ID.

nnn-nnn

Indicates a range of schedule IDs. If the job is scheduled by either of the two IDs specified or any ID between those two IDs, this test is considered true.

(nnn,...,nnn)

Indicates a list of IDs, either specific or ranges, separated by commas and enclosed within parentheses. If any test in the list is true, statements are included or excluded based on the type of statement used.

BD

Indicates a beginning date after which the overrides are in effect.

Size/Type: 5 numeric characters Julian date specified as yyddd
 Default: 00000
 Required: No

BT

Indicates a beginning time-of-day for the BD date.

Size/Type: 4 numeric characters specified as hhmm
 Default: 0000
 Required: No

ED

Indicates an ending date after which the overrides are no longer in effect.

Size/Type: 5 numeric characters Julian date specified as yyddd
 Default: 99999
 Required: No

ET

Indicates an ending time-of-day for the ED date.

Size/Type: 4 numeric characters specified as hhmm
 Default: 2400
 Required: No

JOB

Indicates that inclusion/exclusion is to be based on a matching fully qualified job name.

Size/Type: 1 to 8 alphanumeric characters
 Required: No

RPT

Indicates that inclusion/exclusion is to be based on the presence (Y) or absence (N) of CA-7/RPT.

Default: Y (If SUBSYS=RPT is coded in the DBASE statement in the CA-7 initialization file and CA-7/RPT is active)
 Required: No

OA

Indicates a time-of-day only at which and after which the overrides are to be considered for inclusion or exclusion. The OA time should be lower than the OB time specified.

Size/Type: 4 numeric characters specified as hhmm
 Default: 0000
 Required: No

OB

Indicates a time-of-day only at which and before which the overrides are to be considered for inclusion or exclusion.

Size/Type: 4 numeric characters specified as hhmm
 Default: 2400
 Required: No

CV

Indicates the comparison values against which BD, BT, ED, ET, OA, and OB parameters are to be compared.

Default: DO
 Required: No

DO

Indicates values specified are to be compared against the due out date and the time-of-day values.

DL

Indicates values specified are to be compared against the deadline date and the time-of-day values.

CU

Indicates values specified are to be compared against the current date and the time-of-day values.

The ID, BD, BT, ED, ET, RPT, OA, OB, and CV parameters may be used in combination to accomplish the desired result. At least one of the parameters must be used.

Note: The default date and time used for the test for exclusion or inclusion is the job's due out time unless otherwise specified on the CV parameter. That is, if the due out time for the job falls within the beginning and ending date and time, the statements are excluded or included. The due out time for a demanded job is the current time plus one hour unless otherwise specified on the CV parameter or specified differently with the DOTM and/or LEADTM on the DEMAND command. When using JCL validation to test these statements, the time used is always the current time since due-out and deadline times are not applicable.

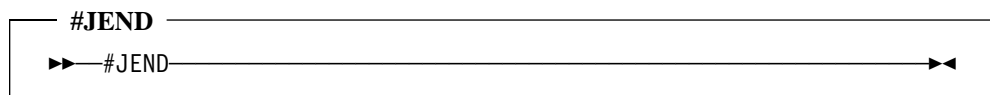
If any errors are encountered in the #JI or #JO override statements (that is, invalid data for ID, BD, BT, ED, ET, RPT, OA, OB, or CV fields), a message is sent notifying the MASTER station that the JCL could not be attached, and the job remains in the request queue in SKELETON status.

The first statement in any JCL member must be a JOB statement and may not be overridden. The exception is a /*PRIORITY statement which may precede the JOB statement. However, #JI and #JO statements must come after the JOB statement.

10.2.2 #JEND Statement

The conditional statements may be terminated by another #JO or #JI statement, an end-of-file, or a #JEND statement. The #JEND statement indicates the end of a set of Scheduled Override control statements.

10.2.2.1 Syntax



10.2.2.2 Usage Notes

There are no other keywords, and it must begin in position 1 of the statement.

Note: #JI, #JO, and #JEND statements are stripped out of the JCL, and overrides applied, as the JCL for the job is brought into the queue. These statements cannot be seen in the JCL for the job in the queue, nor can they be added to the queue JCL.

10.2.3 #XI, #XO, and #XEND Statements

If the user prefers # statements to remain until submission time, the characters #XI, #XO, and #XEND must be substituted for #JI, #JO, and #JEND respectively. The format and field descriptions of the #XI, #XO, and #XEND statements are the same as the #JI, #JO, and #JEND statements. All scheduling parameters have the same meaning when either the #X or #J statements are used. The only difference between #X and #J is that the overrides are applied at submission time for the #X statements. If any errors are encountered in the #XI or #XO override statements at submission time, the job is flushed with an error message.

Any time after the job enters the queues and prior to JCL submission, #X statements can be reviewed and/or changed with the QM.5 screen.

10.2.4 Scheduled Override Examples

Example 1: To omit statements whenever a job is scheduled by Schedule ID 6, use:

```
#JO,ID=6
.
. (JCL statements to be omitted)
.
#JEND
```

Example 2: To include statements until date 00265 at 8.00 AM, use:

```
#JI,ED=00265,ET=0800
.
. (JCL statements to be included)
.
#JEND
```

Example 3:

To omit statements from 00200 to 00206, use:

```
#JO,BD=00200,ED=00206
.
. (JCL statements to be omitted)
.
#JEND
```

Example 4: To include statements from 00200 to 00206, use:

```
#JI,BD=00200,ED=00206
.
. (JCL statements to be included)
.
#JEND
```

Example 5: To include statements whenever a job is scheduled by Schedule IDs 1, 4, 5, 6, and 9, use:

```
#JI,ID=(1,4-6,9)
.
. (JCL statements to be included)
.
#JEND
```

Example 6: To include statements only between 10:00 AM and 4:00 PM, use:

```
#JI,OA=1000,OB=1600  
.  
.(JCL statements to be included)  
.  
#JEND
```

Example 7: To omit statements only between 4:00 PM and 10:00 PM for schedule IDs 9, 10, and 11, use:

```
#JO,ID=9-11,OA=1600,OB=2200  
.  
.(JCL statements to be omitted)  
.  
#JEND
```

Example 8: To include statements only if CA-7/RPT is active, use:

```
#JI,RPT=Y  
.  
.(JCL statements to be included)  
.  
#JEND
```

Note: The JCLxx text editor command or the LJCK command may be used to simulate the JCL inclusions or omissions. This can help test the conditional JCL statements for proper generation. See Edit Command Descriptions on page 11-11. Also see 10.4, “SASSJCLU - JCL Utility” on page 10-24 which can be used to remove expired override statements from PDS libraries.

10.3 Additional Override Statements

CA-7 also provides a number of other functions which can be accomplished with statements which also begin with a # in position 1. These statements may be located anywhere within the execution JCL after the JOB statement. (PROCLIB members containing these statements are not seen by CA-7 and including them there would thus serve no purpose.) They may be scheduled with #Jx statements if desired; however, they may be included by themselves if conditional inclusion is not necessary. Only the #MSG statement may be scheduled with the #Xx statement.

The following statements may be placed in the execution JCL:

- #ARF** Overrides the ARFSET designation on the DB.1 screen when the JCL is attached for the job.
- #HLD** Places the job in HOLD. Similar to the HOLD command and the DB.1 screen HOLD option.
- #JCL** Sets a JCL override requirement.
- #MNT** Considers this run a maintenance type run. Similar to the DB.1 screen MAINT option.
- #MSG** Sends messages to the master station at job submission time. See 10.3.2, “#MSG Statement” on page 10-16.
- #NOX** Makes this run nonexecutable. Similar to the DB.1 screen EXEC option.
- #NTR** Turns off triggering by successful job completion for this run. Similar to the DEMAND command with SET=NTR option.
- #RES** Changes any Workload Balancing resource requirement for this run. Similar to the RESCHNG command. See 10.3.3, “#RES Statement” on page 10-17 for a discussion on the format and keywords.
- #SCC** Defines step-level condition code checking criteria. See the discussion of 10.3.4, “#SCC Statement” on page 10-18.
- #VER** Sets manual verification requirement. Similar to the VERIFY command and the DB.1 screen VERIFY option.
- #7UNI** Flags JCL to be used to send work to Unicenter TNG. It must be the first statement in the member and prevents edits from CA-7 unless Mixed Case Editor support has been enabled.

10.3.1 #ARF Statement

Use the #ARF statement to override the ARFSET designation on the DB.1 screen when the JCL is attached for the job.

10.3.1.1 Syntax

```
#ARF
  ►—#ARF,SET=nnnnnnnn—◄
```

Where:

#ARF

Identifies the statement as an ARFSET override. #ARF must begin in position 1 of the record.

SET

Identifies the name of the ARFSET that is to be used for this run of the job.

10.3.1.2 Example

Suppose that the DB.1 screen for job A designates an ARFSET named ARECOVER to be used for ARF recovery.

The following example illustrates the use of the #ARF statement to override the ARFSET designation:

```
#ARF,SET=ABC
```

In this example, ARFSET ABC will be used to determine ARF recovery for this job instead of ARECOVER.

10.3.1.3 Usage Notes

When a job initially enters the request queue, CA-7 determines whether ARF is to monitor the job. CA-7 selects an ARFSET to be used for monitoring and recovery based on the following hierarchy:

1. If an ARFSET is supplied on the DEMAND or RUN command, it will be used.
2. If no ARFSET is supplied on a DEMAND or RUN command for the job, then the ARFSET named on the appropriate #ARF statement in the JCL will be used.
3. If no ARFSET is supplied on a DEMAND or RUN command for the job, and if no ARFSET is named on a #ARF statement in the JCL, then the ARFSET named on the DB.1. screen will be used.
4. If none of the above sources supplies an ARFSET reference, then the job will not be monitored by ARF.

10.3.2 #MSG Statement

#MSG statements may be used to send messages to the CA-7 Master Station at job submission time. These statements must begin in column 1 and are placed in execution JCL members on the JCL libraries. The edit facility, with the JCL command of database maintenance, can be used to add these statements. The statements may appear within scheduled overrides, in which case sending the messages is conditional.

10.3.2.1 Syntax

<pre>#MSG ▶▶#MSG,message-text▶▶</pre>

10.3.2.2 Usage Notes

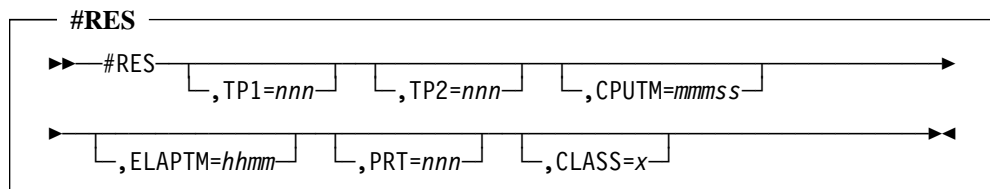
The message-text in columns 6 through 72 is sent immediately preceding the submit message. The actual #MSG statements are never included in the submitted JCL.

Note: The same rules for other # statements also apply to the #MSG statement.

10.3.3 #RES Statement

You may place this statement in the execution JCL to make temporary changes to the high-water workload balancing resource requirements of the job.

10.3.3.1 Syntax



Where:

TP1

Indicates the number of tape drives of TYPE1 that are required of the job.

Size/Type: 1 to 3 numeric characters from 0 to 255

Required: No

TP2

Indicates the number of tape drives of TYPE2 that are required of the job.

Size/Type: 1 to 3 numeric characters from 0 to 255

Required: No

CPUTM

Indicates the amount of CPU time the job consumes in minutes and seconds.

Size/Type: 2 to 5 numeric characters specified as mmmss where mmm may be from 0 to 999; seconds from 0 to 59

Required: No

ELAPTM

Indicates the elapsed time required for the job.

Size/Type: 2 to 5 numeric characters specified as hhmm where hh can be from 0 to 24; mm from 0 to 59

Required: No

PRT

Indicates the initial priority value for the job.

Size/Type: 1 to 3 numeric characters from 0 to 255

Required: No

CLASS

Indicates the WLB job class of the job.

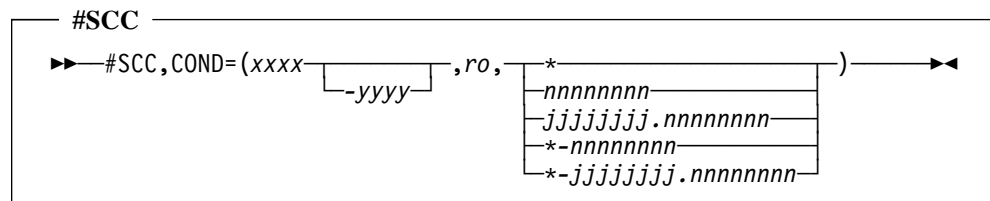
Size/Type: 1 alphanumeric character

Required: No

10.3.4 #SCC Statement

These statements may be placed in the execution JCL to define step level condition code tests used to determine whether the job completed successfully.

10.3.4.1 Syntax



Where:

#SCC

Identifies the statement as a step level condition code test. #SCC must begin in position 1 of the record.

COND

Identifies the following information as condition code test values.

xxxx

Identifies the condition code value to be tested against the actual value returned at execution time. The entry must be numeric and fall within the range of 0 to 4095.

A range of condition codes may be specified as xxxx-yyyy where xxxx is the minimum in the range of condition code values to be tested and yyy is the maximum in the range of condition codes to be tested. Each entry must be numeric and fall within the range of 0 to 4095.

ro

Indicates the relational operator specifying the comparison to be made. If the condition code value returned has the relationship defined by the ro entry, the step value is unacceptable and the job is considered abnormally completed. For example, if ro is set to LT and xxxx is set to 8, the job is marked as completing abnormally if 8 is less than the return code from the step whose name matches the name in the nnnnnnnn parameter. The value must be one of the following:

GT Greater than

GE Greater than or equal to

EQ Equal to

LT Less than

LE Less than or equal to

NE Not equal to

FL Mark the job abnormally terminated if the named step was flushed

Note: Condition code values coded on the #SCC statement are ignored if an RO of FL is specified.

nnnnnnnn jjjjjjj.nnnnnnnn

Indicates the step name (and optional procstepname) of the step(s) to be tested. An * (asterisk) entry indicates that the test applies to all job steps. A test that is to be applied to all job steps is a global #SCC test. A step may be excluded from a global test by coding the statement using the following format:

```
#SCC,COND=(xxxx,ro,*-nnnnnnnn)
```

A #SCC statement coded in this fashion indicates that the test is to apply to all job steps except the one named by nnnnnnnn.

If specifying a test that applies to a step in a cataloged or in-stream procedure, the jjjjjjj identifies the EXEC statement of the calling job step; nnnnnnnn identifies the EXEC statement of the procedure step that issues the return code to be used in the test. If only nnnnnnnn is specified, the test will be applied to all occurrences of that job step name or procedure step name.

10.3.4.2 Example

The following are examples of the #SCC statement:

```
#SCC,COND=(16,LT,STEP0030)
```

In the above example, if 16 is less than the condition code value returned from STEP0030, the job is considered as having terminated abnormally.

Note: A condition code test defined on a #SCC statement is applied to ALL steps whose name matches nnnnnnnn until all such steps have been tested or until conditions defined by a #SCC test have been met. Steps are tested in order of execution.

It is possible to exclude only a couple of steps from a global #SCC test by coding multiple #SCC statements with the same RO and condition code:

```
#SCC,COND=(0,LT,*-STEP5)
#SCC,COND=(0,LT,*-STEP10)
```

The above example would cause all steps of the job except STEP5 and STEP10 to be tested. If the RO or condition code specified are different, each statement would be evaluated separately.

10.3.4.3 Usage Notes

- #SCC statements may optionally be scheduled using the #JI or #JO techniques. If the #JI or #JO are used, different #SCC statements may be used per SCHID.
- #SCC statements may not be scheduled using #XI or #XO and may not be added to JCL in the queue.
- Multiple #SCC statements are allowed for each job step name.
- A #SCC test for a step name of * (asterisk) may be used by itself to apply to each step in the job. If used in combination with a #SCC for a specific job step, the condition code returned by that step is validated against both #SCC condition code tests.
- The RO value on the DB.1 screen must be set to #S or all #SCC statements are ignored. If job-level condition code testing is desired, the DB.1 screen must be used and the RO value is any other valid value (other than #S). In this way #SCC statements can still be in the JCL for future use, but are ignored.
- Since CA-7 only regains control of a job at job completion time, #SCC tests cannot be used to bypass execution of job steps.
- Failure of any condition code test, either at step level or job level, causes the job to be returned to the request queue and flagged with a restart requirement.
- #SCC statements may be located anywhere in the JCL following the JOB statement, but should be positioned in step execution sequence for user readability.
- Overhead for a job-level condition code test as defined on the DB.1 screen is less than using the #SCC statements and should be considered before implementing #SCC tests. For most jobs, the DB.1 screen option probably can be used.
- SMF step termination records cause a job step to be tested against all #SCC tests defined for that step. No tests are made to ensure that a step exists with the name defined in #SCC statements.

10.3.5 Additional Override Examples

1. The job DUSAZZ01 runs Monday through Friday. Monday through Thursday has been defined as SCHID=1 and Friday as SCHID=2. Manual verification is only required on the Friday run.

```
//DUSAZZ01 JOB ...
#JI,ID=2
#VER
#JEND
.
.
.
```

2. Job DUSAZZ01 runs Monday through Friday. Monday through Thursday has been set up as SCHID=1 and Friday as SCHID=2. Job resources have been set up as follows:

```
Tape drives of TYPE1 = 0
Tape drives of TYPE2 = 0
CPU time (mmss)      = 00130
Elapsed time (hhmm)   = 0012
Class                 = A
```

The preceding resources are needed only from Monday through Thursday. On Friday, the job uses two TYPE1 tape drives and requires two minutes and 40 seconds of CPU time and 20 minutes of elapsed time. To change the resource requirements only for the Friday run, enter:

```
//DUSAZZ01 JOB ...
#JI,ID=2
#RES,TP1=2,CPUTM=00240,ELAPTM=0020
#JEND
.
.
.
```

3. Assume job XYZ runs daily, Monday through Friday. But on Friday, a control statement must be included to indicate a week-ending run and an additional file is to be output. Monday through Thursday is defined as SCHID=1 and Friday is defined as SCHID=2. The following sample JCL is for job XYZ:

```
//XYZ JOB ...
//JOB LIB DD DSN=MY.LOADLIB,DISP=SHR
//STEP1 EXEC PGM=MYPGM
//INPUT DD DSN=INPUT.TRANS,DISP=SHR
//RPTDAILY DD SYSOUT=A
#JI,ID=2
//WKLYOUT DD DSN=WEEK.ENDING.FILE,DISP=(NEW,CATLG)...
#JEND
//SYSIN DD *
#JI,ID=1
DAILY
#JI,ID=2
FRIDAY
#JEND
/*
```

4. A job has 10 steps. On STEP01 any condition code other than 5 should be considered invalid. If the condition code returned from any step other than STEP01 is not equal to 0, the job should be moved back to the request queue and flagged for restart.

```
#SCC,COND=(5,NE,STEP01)
#SCC,COND=(0,NE,*-STEP01)
```

5. A job has 5 steps. On STEP01 any condition codes other than 0, 8, 16, and 32 should be considered invalid. There is no condition code checking on STEP02-STEP04. STEP05 should be considered invalid if the condition code is not 0 or 54.

Note: The range specified in the first test is 7-1, this is considered equivalent to 1-7.

```
#SCC,COND=(7-1,EQ,STEP01)
#SCC,COND=(9-15,EQ,STEP01)
#SCC,COND=(17-31,EQ,STEP01)
#SCC,COND=(32,LT,STEP01)
#SCC,COND=(1-53,EQ,STEP05)
#SCC,COND=(54,LT,STEP05)
```

6. A job has 10 steps. On STEP03 any condition code other than 100 should be considered invalid. Only two condition codes are valid for STEP07: 0 and 100. Any condition other than 0 should be considered invalid on all remaining job steps. The following #SCC statements should be coded to effect the correct tests:

```
#SCC,COND=(100,NE,STEP03)
#SCC,COND=(1-99,EQ,STEP07)
#SCC,COND=(100,LT,STEP07)
#SCC,COND=(0,NE,*-STEP03)
#SCC,COND=(0,NE,*-STEP07)
```

7. The JOBCHECK step MUST complete with a condition code of zero for the job to complete normally. The following #SCC statements should be coded:

```
#SCC,COND=(0,NE,JOBCHECK)
#SCC,COND=(,FL,JOBCHECK)
```

Note: See 10.5, “JCL Validation” on page 10-27 for information about testing override statements.

10.4 SASSJCLU - JCL Utility

SASSJCLU is a batch utility program which provides capabilities to:

- Remove expired CA-7 JCL Scheduled Override control statements (#J and #X statements) and the scheduled JCL from a JCL library member of a PDS.
- Print the contents of all or selected JCL members in a JCL PDS data set.

Note: This utility can only be applied against a PDS JCL library. Members that are updated show CA7JCLU as the ID from the last update and ISPF statistics are updated or created.

The desired function(s) to be performed is indicated through PARM values to the program. A SYSIN data set can be used to identify particular members which are to be listed and/or updated.

The utility requires the following execution JCL:

```
//STEPX   EXEC   PGM=SASSJCLU,PARM='xxxxxxxx....x'
//STEPLIB DD     DSN=ca7.loadlib,DISP=SHR
//SYSPRINT DD     SYSOUT=A
//SYSUT1  DD     DSN=JCL-jcl.dataset.name,DISP=OLD
//SYSIN   DD     *           (optional)
control card           (optional)
/*
```

Where:

PARM='xxxxxxxx....x'

Acceptable values for PARM='xxxxxxxx....x' may be any of the following:

PRINT

Print the JCL members selected by control statements in the SYSIN data set.

CLEAN=nn

Remove all scheduled JCL overrides from JCL members selected for which the ending date (ED) is more than nn days prior to today's date.

PANIC

Remove all scheduled JCL overrides from JCL members selected.

10.4.1 Usage Notes

Usage considerations for PARM options are:

- If PARM is omitted, PRINT is the default.
- CLEAN=nn causes CA-7 JCL Scheduled Overrides to be removed from the JCL library if their ending date (ED) was more than nn days prior to today's date. If CLEAN is specified, the 2-digit nn number must be present.
- If CLEAN and PRINT are both specified, expired Scheduled Overrides are removed and the contents of the selected members are listed. If a control statement is present in the SYSIN data set, only selected members are subject to the CLEAN and PRINT options in the SYSIN data set.
- If CLEAN is specified and PRINT is not, any JCL member with expired Scheduled Overrides removed is listed anyway.
- PANIC causes all Scheduled Overrides to be removed, expired or not. If a SYSIN control statement is specified, only JCL members selected have all Scheduled Overrides removed.

10.4.1.1 Control Statement

The SYSIN DD statement is required only if a control statement is to be supplied. If the SYSIN statement is not used, all members are selected. The control statement format, starting in position 1 is:

<pre>1 NAMEKEY=xxxxxxx</pre>

Where:

xxxxxxx

Is a JCL member name, up to 8 characters, or the first part of a JCL member name (generic selection). Generic selection here does not use an asterisk to indicate the end of the common characters. All JCL members whose first characters match the characters specified are subject to the operations specified by the PARM selection.

Note: Care must be taken if member names with fewer than 8 characters are used. For example, if a member name is 4 characters, and they are the same as the first 4 characters of an 8-character member name, both are considered.

Only one control statement is allowed per each execution of SASSJCLU.

10.4.1.2 JCL

This illustrates sample JCL for the SASSJCLU utility:

```
//STEPX      EXEC  PGM=SASSJCLU,PARM='PRINT,CLEAN=14'  
//STEPLIB    DD    DSN=CA-7.LOADLIB,DISP=SHR  
//SYSPRINT   DD    SYSOUT=A  
//SYSUT1     DD    DSN=CA-7.JCLLIB,DISP=OLD  
//SYSIN      DD    *  
NAMEKEY=DUSA  
/*
```

Figure 10-1. SASSJCLU Sample JCL

This sample JCL is used to scan the JCL library for members with names beginning with DUSA and:

- Print those members starting with DUSA.
- Remove all Scheduled Overrides that expired two or more weeks ago from the DUSA members selected.

10.5 JCL Validation

The CA-7 JCL Validation facilities may be invoked from the editor to check for various syntax errors. Such validation is available only when the native editor is used in Full Edit Mode (FEM). Thus, if you are using the CA-7 TSO/ISPF Interface, these subcommands are *not* honored in the ISPF editor.

CA-7 relies on CA-JCLCheck to handle JCL validation. A limited set of CA-JCLCheck functions is provided as part of the CA-7 base product at no extra cost. However, a more extensive report of errors is available when using the CA-7 interface with the complete version of CA-JCLCheck.

10.6 LOAD Command Processing

Use the LOAD/LOADH command to create or re-create job profile data in the database. Job profiles should agree with the current JCL. Therefore, any changes to the JCL must be resynchronized with the database by LOADING the job. See the RELOAD field in 2.2.1, “Field Descriptions” on page 2-4 as an alternative to this command. Use the LOADH command to indicate that the job is to be entered into the queue in CA-7 **hold** status to allow for manual release at a future time.

Use of this command causes the rest of the job's JCL to be flushed. The job returns to the request queue with a JCL error.

You can find more information on the LOAD/LOADH command in the *CA-7 Commands Guide*.

10.7 Special Override Library

CA-7 supports a JCL override library which is intended to handle any onetime JCL needs that arise.

The combination of the special override library and the various special override statements discussed in 10.2, “Scheduled Overrides” on page 10-6 give the user the ability to handle virtually every special need that may arise. See that discussion for more details on the various special functions which are available.

10.7.1 Defining a Special Override Library

Although PDS, CA-Panvalet, and CA-Librarian libraries are supported for execution JCL for production jobs, this special override library is global for all jobs under the control of CA-7 and must be a PDS. Only one special override library may be defined to CA-7.

This special override library must be allocated by the user and made available to CA-7 through an appropriate DD statement and an initialization file JCL statement which specifies INDEX=254. This library is then available to the user on the various screens and commands as JCL ID of 254.

10.7.2 Creating Temporary JCL

Execution JCL can be placed here, through the DB.7 screen, any time prior to the scheduling into the queues of the job which uses that JCL. The DB.1 screen field USE-OVRD-LIB must also be set to Y for the special override library JCL to be used. If the DB.7 screen is used to store the JCL in the special override library and the member name is the same as the job name, the USE-OVRD-LIB field is automatically set to Y.

The next time the job is scheduled into the queues, CA-7 uses the special override library JCL instead of retrieving the JCL from the primary library specified with the ID field in the JCL section of the DB.1 screen.

10.7.3 Using Temporary JCL

When the job is initially brought into the queues and the USE-OVRD-LIB value is Y, the JCL is copied in from the special override library instead of the normal production library specified with the ID field on the DB.1 screen. The USE-OVRD-LIB value is then automatically reset to a value of N, to prevent that JCL from being used again.

The JCL that was copied in from the special override library is used until a successful completion of this run is accomplished. It may be altered using the QM.5 screen if necessary. When the job completes successfully, the JCL is then deleted from the special override library if the member name DB.1 screen field is the same as the job name.

Note: Your installation may choose to prevent the automatic deletion of JCL from the special override library. The OVJCL keyword in the OPTIONS statement in the CA-7 initialization file can be used to prevent this automatic deletion. If this option is set, it applies to all CA-7 jobs. See the *CA-7 Systems Programmer Guide*, OPTIONS statement for more information. Before using this option, you should read 10.8, “Alternate JCL Libraries” on page 10-31 to see if they are appropriate for your situation.

10.7.4 Other Considerations

A LOAD step automatically sets the USE-OVRD-LIB value to N whenever it is executed. If a job has JCL in the special override library waiting for the job to be scheduled and a permanent change is made to the master JCL in the normal production library, any load of the JCL causes the USE-OVRD-LIB flag to be reset to N. Therefore, such changes must be avoided, or at least not reloaded with a LOAD top line command, until after the job has been scheduled into the queues with the set of JCL from the special override library.

10.8 Alternate JCL Libraries

CA-7 supports a primary/alternate approach to JCL libraries. This enables the user to place temporary JCL in a staging-type of library to be used for more than just a single execution of a job. The special override library, discussed earlier in this chapter, accommodates onetime overrides only and the temporary member is deleted after just one execution of the job. CA-7 LOAD processing does not use the alternate JCL library (when using the LOAD command).

10.8.1 Defining an Alternate Library

In the initialization file, JCL libraries are defined with a JCL statement. In each of these statements, the ALT parameter is used to define which of the defined libraries is to be the alternate library. Only one alternate can be defined for each library.

Libraries may be any of the supported organizations and do not have to be PDS like the global special override library.

10.8.2 Creating Temporary JCL

Temporary JCL can be stored in an alternate library from either CA-7 or from outside of CA-7 (for example, TSO, CA-Roscoe, and so forth). The JCL ID field on the DB.1 screen still indicates the JCL resides in the primary library.

10.8.3 Using Temporary JCL

When CA-7 schedules a job which uses a JCL library having an alternate library defined, CA-7 searches the alternate library first. If the JCL member is found in the alternate library, CA-7 uses the alternate JCL and flags the job queue entry. If the JCL member is not found in the alternate library, CA-7 searches the primary library.

Unlike the special override library previously discussed, no action is taken by CA-7 at job completion to delete JCL used from an alternate library. As long as a JCL member exists in an alternate library, it is used for job scheduling.

The JCL ID specified on the DB.7 screen must specifically request an alternate library since the DB.1 screen does not indicate alternate usage. Likewise, the LJCL command shows JCL only from the primary library.

Runtime messages to the Master Station and the CA-7 comment statement included with the JCL for the job reflect the JCL ID number for the library from which the JCL was taken for execution.

10.8.4 Other Considerations

Unlike the onetime Override library (INDEX=254), which is automatically reset by CA-7, the user is responsible for ensuring deletion of temporary JCL from the alternate library once it is no longer needed.

One library can be defined as the alternate for more than one primary library.

An alternate library can have its own alternate. However, searching is only done for the primary library requested and its single alternate.

Chapter 11. Edit Facility

The CA-7 edit facility is an interactive component used to create, modify and manage 80-character, card-image data. With this facility, job streams (JCL) can be created or changed prior to submittal and documentation (prose) can be maintained in the CA-7 database.

This facility consists of files, screens, programs, and a command. The command is EDIT which can be issued as a top line command or as a screen function. The files are referred to as active area and the edit work file (EWF). The screens and programs vary based on the environment from which EDIT is issued.

When this facility is invoked, the active area may be cleared or loaded with source data, if requested. The active area is copied to the EWF and an editor program is executed to make the desired changes. When editing is complete, any saved data is copied back to the active area from where it can be copied to the source file or some other target file. Figure 11-1 on page 11-2 shows the edit facility data flow.

SCREENS/ COMMANDS

DB.7 (JCL)
QM5
DB.4 (PROSE)

EDIT

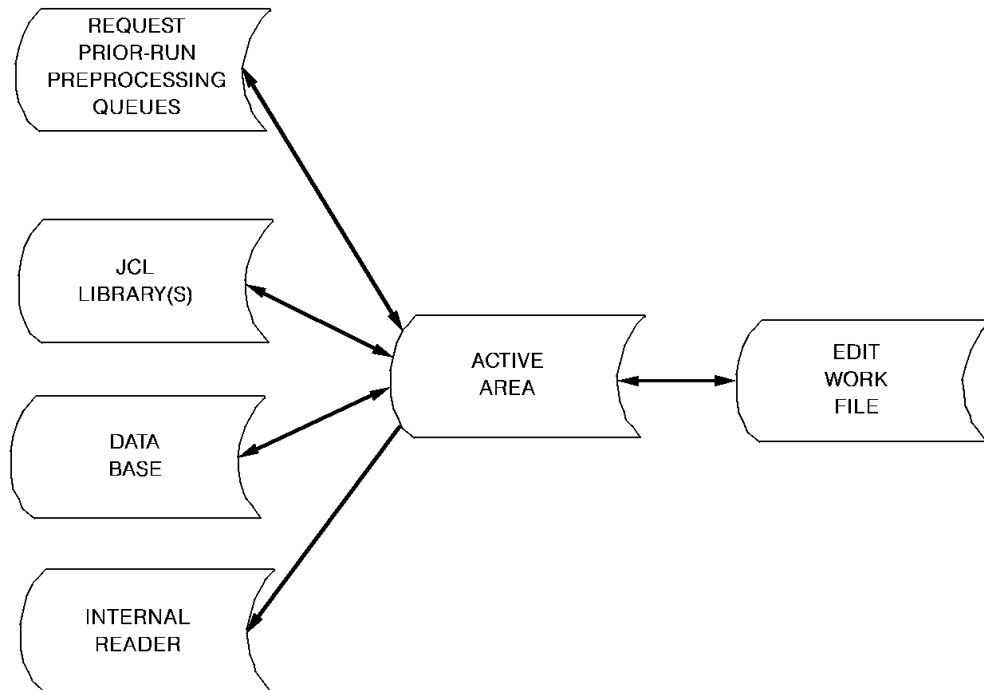


Figure 11-1. Relationship of Commands, Data Sets, and Screens

11.1 Active Area

The active area is a dynamic file which is suballocated from the scratch queue for each terminal user. Allocation is performed one track at a time, as needed, to minimize disk space usage. Maximum allocation is controlled by the SCRTRK or SWFTRK parameter on the DAIO statement in the initialization file. See the *CA-7 Systems Programmer Guide* for more information about these controls.

This file is initialized by the CLEAR and FETCH functions. The CLEAR function removes all data from the file and unallocates its disk space. The FETCH function allocates disk space and copies data into the file. Several of the DB screens show a field called ACTIVE SIZE which tells the terminal user how many lines of text are in this active area. A CLEAR function is performed at every logon and logoff.

11.2 Edit Work File

The edit work file (EWF) is a dynamic file created for each terminal user when EDIT is invoked. For the CA-7 environment, the file is suballocated from the scratch queue as previously described for the active area. For other environments, it is allocated according to the rules and conventions of the environment where requested. The EWF space is released when leaving the editor.

The EWF is initialized by copying the contents of the active area. An editor program is invoked in the terminal user's environment to make changes to the EWF data. When editing is complete, any saved updates are copied back to the active area and the EWF is freed.

11.3 Environments

Although the editing process is very similar, the editor programs used in each environment are very different. Also, there are considerations unique to each environment which are in this chapter. Currently, the edit facility supports the following environments:

- CA-7
- TSO/ISPF

11.4 Using the Editor

There are three general activities involved in using the editor:

- invoking the editor,
- editing text data,
- leaving the editor.

When using CA-7, the editor is usually invoked from another command or screen, such as DB.7 or DB.4, although it can be requested with the EDIT top line command. While under the editor's control, several subcommands are available for editing the text data. After making any desired changes, it is necessary to leave or exit from the editor in order to retain or make use of the updated text.

11.4.1 Invoking the Editor

The editor is invoked by the top line EDIT command or by using the EDIT or FE (Fetch and Edit) functions from database maintenance or queue maintenance screens or commands. Such functions cause the transfer of data from the active area to an edit work file, as described above, then engage the editor programs for the appropriate environment.

11.4.2 Editing Text Data

Once in the editor, text can be created or changed using the subcommands appropriate to the environment where editing occurs. Subcommands for the CA-7 environment are described in 11.5, "CA-7 Text Editor Environment" on page 11-8. Subcommands for the TSO/ISPF environment are described in 11.6, "TSO/ISPF Editor Environment" on page 11-30.

11.4.3 Leaving the Editor

After completing the editing tasks desired, the editor must be exited and changes saved, if desired. When changes are to be saved, the editor must be told to retain the updates and the updated EWF data must be transferred back to the active area. If changes are not to be saved, the editor must be told to release the EWF and not transfer data back to the active area.

For the CA-7 environment, the subcommands to leave the editor and save the changes are SAVE, SS (SAVE-SAVE), and SR (SAVE-REPLACE). The subcommand to leave the editor without retaining EWF changes is EXIT. These subcommands are described in 11.5, “CA-7 Text Editor Environment” on page 11-8.

For the TSO/ISPF environment, ISPF edit macros are provided which correspond to required CA-7 editor subcommands. Also, when the editor is invoked, a default subcommand is set. If changes are not to be saved, the CA7EXIT subcommand must be requested.

11.5 CA-7 Text Editor Environment

The CA-7 text editor routines provide the capability to create or change text information in the CA-7 database, queues, and JCL data sets. These routines operate in the CA-7 environment servicing multiple terminal users from a single address space and using a single set of support files (queues). The editor can be invoked using interactive online screens or batch card-image input.

11.5.1 Edit Modes

Earlier versions of CA-7 contained a basic text editor. This editor is still available, but is generally used by the batch terminal interface. For online editing, a more flexible editor was developed that has a more flexible set of routines. Therefore, two versions or modes of the text editor are available. These modes are referred to as Full Edit Mode (FEM) and Alternate Edit Mode (EDIT or AEM). The Full Edit Mode offers more features than the Alternate Edit Mode.

When EDIT is entered either as a top line command, from the DB Menu Screen or from any formatted screen (other than a DB.2.n screen), FEM is automatically invoked (except from batch). Once in FEM, the RETURN subcommand activates Alternate Edit Mode. To return to Full Edit Mode from Alternate Edit Mode, enter the FEM subcommand.

11.5.1.1 Usage Considerations

- To terminate the text editor modes without retaining changes, enter the EXIT subcommand.
- FEM is **not** a top line command. It is only used in the text editor facility.
- If the text editor was entered from a top line EDIT command or from a formatted screen where EDIT is **not** a listed function (for example, DB.1, DB.6), return is to the DB Menu screen.
- If the text editor was entered from a formatted screen (other than DB.2.n) where EDIT is a valid function (for example, DB.7, DB.4), the return is to the originating screen.
- Unprintable (or hex) data can cause a 3270 screen error and terminate the user session.

11.5.1.2 Full Edit Mode (FEM)

Full Edit Mode is entered through the EDIT command, by the FE (Fetch Edit) function, or through the FEM command if already in the Alternate Edit Mode. The FEM command can only be entered as a command from the Alternate Edit Mode. FEM reduces the number of operations in editing data. It also allows greater text change flexibility.

Normally, when FEM is entered, the edit work file (EWF) is positioned at the top. It is possible, however, when entering FEM from the Alternate Edit Mode, to specify a starting sequence number. For example, FEM 5725 would place the user in Full Edit Mode beginning at the line (or next highest) whose sequence number is 5725.

FEM Screen Areas: A Full Edit Mode (FEM) screen consists of six areas of information:

1. Scale Line
2. Status Information
3. Data Area
4. Command Area
5. Message Line
6. Sequence Numbers/Subcommands

For examples of the screens, see "FEM Sample Screens" on page 11-19.

The first area, containing the scale line, appears at the top and bottom of the display. This allows for column-oriented changes in the data area. (An asterisk denotes the last effective column.)

The second area contains status information. It appears on the right half of line 2. Items of information found here are FILL/NOFILL, MIXED/UPPER case setting, insert increment I (010) and NUM/NONUM.

The third area is the active area. It contains information residing in the edit work file (EWF).

The fourth area on the FEM screen, the command line, appears on the left half of line 2. It is from this line that the commands are input. When the text editor is first entered and the EWF is empty, lines must be inserted with an INSERTnnn command to enable input to the EWF.

The fifth area is the message line on the last screen display line. When there is an error or specific status is required, this line is used. Under normal processing, this line remains blank.

The sixth area on the FEM screen contains the sequence numbers of the data lines being displayed. This area is used as a subcommand area to delete lines, insert blank lines and position the display. More than one subcommand can appear in a single sequence number.

The following rules apply when the data area, subcommand (sequence number field) and command lines are used and the Enter key is pressed:

- When there are no changes to the data area, and no subcommand or command is entered, the display is paged forward (PF) or backward (PB) depending upon the default page command showing in the command line.
- When there are changes to the data area, and no subcommand or command is entered, the data changes are applied and the display is positioned according to the default page command (PF or PB).
- When subcommands are entered and no command is entered, data area changes are made, subcommands are processed and the display is positioned at the last P (Position to this line) subcommand on the screen, or the first line of the current display if no P subcommand was specified.
- When a command is entered, all subcommands and data area changes are processed. If the command is a display positioning command (TOP, BOTTOM, MD, PF, PB, HF, HB, and so forth), positioning with the subcommand P is ignored. If the command is not a display positioning command, the last P subcommand is honored, or the first line of the current display designates the position if no P subcommand is specified.

FEM Subcommands: The following subcommands can be placed in the sequence number area:

P

Positions the display to this line. The last P found on the page is honored.

D

Deletes the line at this sequence number from the EWF (edit work file).

DD

Deletes this line and all subsequent lines to the next DD or end of page.

Note: A DD in the first sequence field of the page when there are no other DDs coded deletes all lines on the current page from the EWF once Enter is pressed.

Innn

Inserts nnn blank lines after the line where the I appears. nnn must be from 1 to 255. Leading zeros are not required. The number of blank lines (nnn) to be inserted **must be truncated** by a blank, another subcommand (I, P, D, DD, or R), the EOF key (erase to End of Field) or the end of the sequence field. These blank lines physically exist in the EWF and may be altered by making changes in the data area of subsequent displays. If not truncated, the sequence numbers remaining are considered part of the lines to be inserted.

Rnnn

Repeats nnn lines after the line where the R appears. The line where the R appears is repeated. nnn must be from 1 to 255 and **must be truncated** as in Innn.

Note: Errors in subcommands cause the subcommand to be ignored with no error notification.

Editor Command Descriptions: The following details the commands available under the CA-7 text editor. (The short form of the command is underlined.)

BOTTOM

Positions the FEM display to the bottom of the EWF. Available in FEM only.

BREAK [nn]

Changes the maximum number of lines to be displayed when the EWF is listed.

Value of nn can be from 00 to 99. The default is 99 lines. BREAK with no parameter establishes the default. Available in FEM and AEM.

CLEAR

Deletes all lines in the edit work file (EWF). Lines of information added by the EDIT commands, or those already present in the Active Area when the text editor was entered, are deleted. Available in FEM and AEM.

COL [n,m]

Displays only columns n through m for data area changes. The column where the asterisk appears on the scale line is the last affected column. The maximum number of columns displayed is 72. COL specified with no parameters sets the default columns of 1 through 72. Available in FEM only.

COPY i[,m,n,t]

Duplicates lines m through n and places them after line i, t number of times. When n is omitted, line m through the last line are copied t number of times. Two commas must separate the m and t values when n is omitted. When both m and n are omitted, the last line of the edit work file (EWF) is copied t number of times and three commas must precede the t. When t is omitted, a default of 1 is assumed. Sequence numbers are assigned to the inserted lines using the current system increment value (see INSERT). The first line copied or moved after the insert line (i) is assigned a sequence number of i plus the current system increment value. All subsequent inserted lines are renumbered using the system increment value. When many new lines are inserted, existing lines following the insert point may be renumbered (using the current system increment) to ensure that the EWF remains in sequence. Available in FEM and AEM.

DELETE m[,n]

Removes lines from the EWF. When n is omitted, only line m is deleted. When both m and n are specified, lines m through n are deleted. Available in AEM.

EDIT

Manipulates specified character strings of data, lines of data or columns for reentry into the EWF. The following EDIT commands are discussed separately. These are only available in AEM.

EDIT LIST [m,n]

Enters Edit List mode. Lines displayed may be changed by keying the changes into the EWF and pressing Enter. A line displayed on the screen may be deleted by keying \$EDEL in the first 5 characters, or by positioning the cursor at the first character of the line and pressing the EOF key (erase to End-of-Field). Screens are returned until either the end of the EWF is reached, line n is displayed, or \$EEND is keyed on the last line. (\$EEND may be keyed on any line to exit Edit List mode. Any lines displayed from the \$EEND to the end of the current Edit List output page, however, are deleted from the EWF.)

EDIT /string/[m,n]

Searches the EWF for the string specified. Slash (/) can be any special character and is used as the string delimiter. All lines found from m through n with the specified string of characters are displayed at the terminal. When n is omitted, only line m is scanned. When both m and n are omitted, the entire EWF is scanned. The maximum number of characters for the string is 54.

EDIT /string1/string2/[m,n]

Replaces character string1 with character string2. All occurrences of string1 from line m through n are replaced by string2. When n is omitted, only line m is scanned for string1. When both m and n are omitted, the entire EWF is scanned. The maximum number of characters for string1 and string2 is 54.

EDIT /string//[m,n]

Deletes a string of characters. All occurrences of the string from line m through n are deleted. When n is omitted, only line m is scanned for the string. When both m and n are omitted, the entire EWF is scanned. The maximum number of characters for the string is 54.

EDIT //string/[m,n]

Replaces the contents of the edit columns with the string, regardless of the current character content. For all lines m through n, the current edit columns are spaced out and the string is left-justified into this area. When n is omitted, only line m is changed. When both m and n are omitted, the entire EWF is changed. The maximum number of characters for the string is 54.

EDIT x[,y]

Changes the start and end columns to be scanned by the other EDIT commands. When y is omitted, x is the new end column. When x and y are both specified, x is the new beginning column and y is the new end column. When y is specified, x must always be less than or equal to y. The values of x and y must be from 1 to 80.

The following are examples of this command:

EDIT 73,80	Changes scan columns to be columns 73 through 80.
'E //SASEEDIT/'	Fills columns 73 through 80 with SASEEDIT.
E 1,72	Resets scan columns.

EXIT

Returns to the original function that invoked the text editor. The contents of the active area are not replaced by the edited EWF. This command is used to exit the text editor without changing the contents of the active area. See SAVE for changing the active area contents. Available in FEM and AEM.

FEM [n]

Used to enter Full Edit Mode (FEM). n is the sequence of the first line of information to be positioned in the display. If n is omitted, then the top of the EWF is assumed.

FILL

Indicates trailing spaces in the data area of the display are to be preserved. Data area changes made by positioning to the end of displayed lines are not left-justified to the first nonblank character on the line. Available in FEM only.

FIND /xxx...x/[n,m]

Searches for a given string of characters (xxx...x) and positions the display at the next line containing those characters. The slash (/) characters are the string delimiters and can be any special characters. Searching is always forward from the current line. To initiate the search from the beginning of the display, first enter the TOP command and then the FIND command. FIND always initiates the search for the next occurrence of the character string. The n and m limits the search to between lines n through m of the current command. If m is omitted, only line n is searched if it follows the top line of the display. If n and m are omitted, the EWF is searched forward from the current position. The string xxx...x is retained. Subsequent FIND commands with no operands search forward for the last string of characters specified by FIND. Available in FEM only. The following is an example of this command.

FIND /ABC/0,99999	Searches forward after the first line displayed for the next line containing ABC. A subsequent FIND positions the display to the next line containing ABC. If ABC is not found after searching forward, the display is positioned at the top. A subsequent FIND then searches for ABC in any following line. Available in FEM only.
-------------------	---

HB n]

Positions the display n half-pages backward. If omitted, n is assumed to be 1. Available in FEM only.

HF n]

Positions the display n half-pages forward. If omitted, n is assumed to be 1. Available in FEM only.

INSERT [m,i]

Used to request Data Insert processing. The operator fills a page with input lines and presses Enter. The lines are then added to the EWF and a new blank screen with continued sequence numbers is presented for more input. To terminate Data Insert processing, either key in \$IEND after the last line to be inserted, or enter a null line after the last line to be inserted.

When both m and i are specified, Data Insert mode processing adds lines beginning with line m, incremented by i. When i is omitted, m changes the system increment value for the INSERT, MOVE and COPY commands. The default of i is 10. If line m exists in the EWF, Insert mode is established at line m + i (lines are inserted after line m). Line m is not altered. If both m and i are omitted, Insert mode is established at the end of the EWF. (In this case, a comma must be used to indicate the absence of m and i, if sequence numbers are in 73-80.) The first inserted line has a sequence number consisting of the last sequence number in the file plus the current edit increment. The increment value specified here is reflected in the status information. The INSERT command may be requested in either AEM or FEM. If entered in FEM, control is passed to the editor in AEM for the duration of Data Insert processing, and is returned to the editor in FEM when processing is complete.

Note: Mixed case support is available only in FEM. Hence, all data added during Data Insert processing will be translated to uppercase.

JCL nnn[,yyddd,hhmm]

CA-Driver procedures are expanded, if used. Then CA-7 scans the EWF and reports syntax errors. The format of the report depends on whether the default or the complete version of CA-JCLCheck is used. After the report is displayed, control is returned to the editor environment where the JCL subcommand was invoked. Default values for schedule ID, date and time are used in CA-Driver procedures and in #JI/#JO statements unless those values are overridden. Use nnn to override the default SCHID (1). The default date (current date) may be overridden using yyddd. Use hhmm to override the default time (current time).

JCLL nnn[,yyddd,hhmm]

CA-Driver procedures are expanded, if used. Then CA-7 scans the EWF and lists the JCL statements along with any syntax errors detected. The format of the report depends on whether the default or the complete version of CA-JCLCheck is used. After the report is displayed, control is returned to the editor environment where the JCLL subcommand was invoked. Default values for schedule ID, date and time are used in CA-Driver procedures and in #JI/#JO statements unless those values are overridden. Use nnn to override the default SCHID (1). The default date (current date) may be overridden using yyddd. Use hhmm to override the default time (current time).

JCLS nnn[,yyddd,hhmm]

CA-Driver procedures are expanded, if used. Then CA-7 scans the EWF and reports syntax errors. The format of the report depends on whether the default or the complete version of CA-JCLCheck is used. A SAVE subcommand is processed if no errors are detected and control is returned to the point where the editor was invoked. If errors are detected then after the report is displayed, control will return to the editor environment where the JCLS subcommand was issued. Default values for schedule ID, date and time are used in CA-Driver procedures and in #JI/#JO statements unless those values are overridden. Use nnn to override the default SCHID (1). The default date (current date) may be overridden using yyddd. Use hhmm to override the default time (current time).

JCLSR nnn[,yyddd,hhmm]

CA-Driver procedures are expanded, if used. Then CA-7 scans the EWF and reports syntax errors. The format of the report depends on whether the default or the complete version of CA-JCLCheck is used. An SR subcommand is processed if no errors are detected and control is returned to the point where the editor was invoked. If errors are detected then after the report is displayed, control will return to the editor environment where the JCLSR subcommand was issued. Default values for schedule ID, date and time are used in CA-Driver procedures and in #JI/#JO statements unless those values are overridden. Use nnn to override the default SCHID (1). The default date (current date) may be overridden using yyddd. Use hhmm to override the default time (current time).

JCLSS nnn[,yyddd,hhmm]

CA-Driver procedures are expanded, if used. Then CA-7 scans the EWF and reports syntax errors. The format of the report depends on whether the default or the complete version of CA-JCLCheck is used. An SS subcommand is processed if no errors are detected and control is returned to the point where the editor was invoked. If errors are detected then after the report is displayed, control will return to the editor environment where the JCLSS subcommand was issued. Default values for schedule ID, date and time are used in CA-Driver procedures and in #JI/#JO statements unless those values are overridden. Use nnn to override the default SCHID (1). The default date (current date) may be overridden using yyddd. Use hhmm to override the default time (current time).

LIST [m,n]

Displays the contents of the EWF. When n is omitted, only line m is displayed. When both m and n are specified, all lines from m through n are displayed within the limits of the current BREAK value. When m and n are both omitted, the entire EWF is listed within the limits of the current BREAK value. Available in AEM only.

MD [n]

Moves the display to the sequence number specified by n. If n is 0 or omitted, MD is equivalent to the TOP command. If n is 99999, MD is equivalent to the BOTTOM command. Available in FEM only.

MIXED

Changes the current case setting to mixed case. If the current case setting is 'MIXED' then characters will not be translated, thus allowing both upper- and lower-case characters to be entered. See the discussion of the UPPER command in this section for more information on changing the case setting.

The initial case setting is 'UPPER' unless lowercase characters are detected. If lowercase characters are detected and INITCASE=Y is specified on the OPTIONS statement in the CA-7 initialization file, the initial case setting is 'MIXED'.

This subcommand is not valid unless INITCASE=Y is specified on the OPTIONS statement in the CA-7 initialization file. Available in FEM only.

MOVE i[,m,n,t]

Places lines m through n after line i, t number of times. When n is omitted, m is the only line moved. When both m and n are omitted, the last line of the EWF is moved. When t is omitted, a default of 1 is assumed. Sequence numbers are assigned to the inserted lines using the current system increment value (see INSERT). The first line copied or moved after the insert line i is assigned a sequence number of i plus the current system increment value. All subsequent inserted lines are renumbered using the system increment value. When many new lines are inserted, existing lines following the insert point may be renumbered (using the current system increment) to ensure that the EWF remains in sequence. Available in AEM and FEM.

Note: When m or n are omitted, commas must be specified to denote the omission of these positional parameters.

NOFILL

Indicates that trailing spaces in the data area of the display are not preserved. Data area changes made by positioning to the end of displayed lines are left-justified to the first nonblank character of the line. Spaces must be inserted to hold position. Available in FEM only.

NONUM

Indicates that sequence numbers are external to the data. (Same as the XSEQ.) Available in FEM only.

NUM

Indicates that sequence numbers are to be found in lines 76 through 80 of the data. (Same as XSEQ OFF.) Available in FEM only.

PB [n]

Positions the display n pages backward. If omitted, n is assumed to be 1. Available in FEM only.

PF [n]

Positions the display n pages forward. If omitted, n is assumed to be 1. Available in FEM only.

RENUM m,i

Renums the entire EWF. Both m and i are required. m is the number to be assigned to the first line of the EWF. i is the increment for each succeeding line number. Available in FEM and AEM.

RETURN

Used to enter the Alternate Edit Mode from FEM.

SAVE

Used to return to the original function that invoked the Text Editor. The contents of the active area are replaced by the current EWF. This command is used to exit the text editor with the edited EWF replacing the active area. Available in FEM and AEM.

SCALE [OFF]

Indicates whether a column indicator heading is to be displayed on each output page. SCALE OFF turns off Scale mode. SCALE with no operand displays the column scale. Available in AEM only.

SR

Performs two functions, SAVE and REPL. The current EWF replaces the active area, control returns to the screen which was used to enter the text editor, and a REPL function is performed. Available in FEM and AEM.

Note: If JCL is being replaced in the Request queue, the SR command also sets an outstanding JCL override requirement off.

SS

Performs two functions, SAVE and SAVE. The current EWF replaces the active area, control returns to the screen used to enter the Text Editor, and a SAVE function is performed. Available in FEM and AEM.

STATUS

Displays current information about the EWF. This information consists of the number of lines in the EWF, last line number, edit scan columns, and so on. The **PA1** key must be pressed (page forward) to redisplay the EDIT screen after a STATUS command. Available in FEM and AEM.

TOP

Positions the display to the top of the EWF. Available in FEM only.

TRACE [EDIT] | [OFF]

Sets on (TRACE EDIT) or sets off (TRACE OFF) the tracing of lines altered by the editor commands. With TRACE on, each line changed by the EDIT command is displayed. (BREAK does not apply.) Available in FEM and AEM.

UPPER

Changes the current case setting to uppercase. If the current case setting is 'UPPER' then all characters will be translated to uppercase.

The initial case setting is 'UPPER' unless lowercase characters are detected. If lowercase characters are detected and INITCASE=Y is specified on the OPTIONS statement in the CA-7 initialization file then the initial case setting is 'MIXED'.

XSEQ [OFF]

Establishes External Sequence mode. XSEQ alone indicates that the sequence numbers are to be external to the 1- to 80-character line image. XSEQ OFF indicates that columns 76 through 80 of the line image are to be the sequence number field. When an active area is saved and XSEQ is in effect, the contents of columns 76 through 80 are the same values present when the text editor was entered. When XSEQ OFF is in effect and an active area is saved, the current line number for each line replaces columns 76 through 80 of the line. Available in FEM and AEM.

FEM Sample Screens: The following are FEM sample screens. Each screen displays the use of subcommands in the sequence field areas or changes made to the data area.

The following screen shows five of the six areas discussed previously. The last area of the screen, the message line, is blank.

FEM Sample Screen 1

```

      ----+----10---+----20---+----30---+----40---+----50---+----60---+----70*
PF                                     NOFILL I(010)
00010 //DUSAXX01 JOB HE67YFSH,ACPAY,REGION=40K,TIME=0003,CLASS=A
00020 /*ROUTE PRINT RMT1
00030 //STEP1 EXEC PGM=IEBGENER
00040 #JI,ID=1
00050 #MSG,*****
00060 #MSG,*                NETWORK RUN AS SCHID OF 1                *
00070 #MSG,*****
00080 #JI,ID=2
00090 #MSG,*****
00100 #MSG,*                NETWORK RUN AS SCHID OF 2                *
00110 #MSG,*****
00120 #JI,ID=3
00130 #MSG,*****
00140 #MSG,*                NETWORK RUN AS SCHID OF 3                *
00150 #MSG,*****
00160 #JI,ID=4
00170 #MSG,*****
00180 #MSG,*                NETWORK RUN AS SCHID OF 4                *
00190 #MSG,*****
00200 #JEND
      ----+----10---+----20---+----30---+----40---+----50---+----60---+----70*

```

The following screen reflects a data area change on line 60 of FEM Sample Screen 1 on page 11-19. Changes are made by keying over existing line data and then pressing Enter.

FEM Sample Screen 2

```

-----+-----10-----+-----20-----+-----30-----+-----40-----+-----50-----+-----60-----+-----70*
PF                                           NOFILL I(010)
00010 //DUSAXX01 JOB HE67YFSH,ACPAY,REGION=40K,TIME=0003,CLASS=A
00020 /*ROUTE PRINT RMT1
00030 //STEP1 EXEC PGM=IEBGENER
00040 #JI,ID=1
00050 #MSG,*****
00060 #MSG,*                THIS IS A CHANGE EXAMPLE                *
00070 #MSG,*****
00080 #JI,ID=2
00090 #MSG,*****
00100 #MSG,*                NETWORK RUN AS SCHID OF 2                *
00110 #MSG,*****
00120 #JI,ID=3
00130 #MSG,*****
00140 #MSG,*                NETWORK RUN AS SCHID OF 3                *
00150 #MSG,*****
00160 #JI,ID=4
00170 #MSG,*****
00180 #MSG,*                NETWORK RUN AS SCHID OF 4                *
00190 #MSG,*****
00200 #JEND
-----+-----10-----+-----20-----+-----30-----+-----40-----+-----50-----+-----60-----+-----70*

```

The following screen reflects the request for deletion of three lines from the EWF. You may place a D in any position of the sequence field to delete the line. See FEM Sample Screen 4 on page 11-22 for the results of this edit.

FEM Sample Screen 3

```

-----+-----10-----+-----20-----+-----30-----+-----40-----+-----50-----+-----60-----+-----70*
PF                                           NOFILL I(010)
00010 //DUSAXX01 JOB HE67YFSH,ACPAY,REGION=40K,TIME=0003,CLASS=A
00020 /*ROUTE PRINT RMT1
00030 //STEP1 EXEC PGM=IEBGENER
00040 #JI,ID=1
00050 #MSG,*****
00060 #MSG,*                THIS IS A CHANGE EXAMPLE                *
00070 #MSG,*****
00080 #JI,ID=2
00090 #MSG,*****
00100 #MSG,*                NETWORK RUN AS SCHID OF 2                *
00110 #MSG,*****
00120 #JI,ID=3
00130 #MSG,*****
00140 #MSG,*                NETWORK RUN AS SCHID OF 3                *
00150 #MSG,*****
00160 #JI,ID=4
00170 #MSG,*****
00180 #MSG,*                NETWORK RUN AS SCHID OF 4                *
00190 #MSG,*****
00200 #JEND
-----+-----10-----+-----20-----+-----30-----+-----40-----+-----50-----+-----60-----+-----70*

```

The following screen reflects the results of the edit performed in FEM Sample Screen 3 on page 11-21 deleting three lines: 00080, 00120, and 00160.

Now a P is placed in the sequence field on line 100. See FEM Sample Screen 5 on page 11-23 for the results.

FEM Sample Screen 4

```

      ----+-----10---+-----20---+-----30---+-----40---+-----50---+-----60---+-----70*
PF                                     NOFILL I(010)
00010 //DUSAXX01 JOB HE67YFSH,ACPAY,REGION=40K,TIME=0003,CLASS=A
00020 /*ROUTE PRINT RMT1
00030 //STEP1 EXEC PGM=IEBGENER
00040 #JI,ID=1
00050 #MSG,*****
00060 #MSG,*                THIS IS A CHANGE EXAMPLE                *
00070 #MSG,*****
00090 #MSG,*****
P0100 #MSG,*                NETWORK RUN AS SCHID OF 2
00110 #MSG,*****
00130 #MSG,*****
00140 #MSG,*                NETWORK RUN AS SCHID OF 3                *
00150 #MSG,*****
00170 #MSG,*****
00180 #MSG,*                NETWORK RUN AS SCHID OF 4                *
00190 #MSG,*****
00200 #JEND
      ----+-----10---+-----20---+-----30---+-----40---+-----50---+-----60---+-----70*

```

The following screen reflects the result of the P subcommand to position the display from FEM Sample Screen 4 on page 11-22 to line 100.

Now an I3 is placed in sequence line 200. See FEM Sample Screen 6 on page 11-24 for the results.

FEM Sample Screen 5

```

-----+-----10-----+-----20-----+-----30-----+-----40-----+-----50-----+-----60-----+-----70*
PF                                           NOFILL I(010)
00100 #MSG,*                               NETWORK RUN AS SCHID OF 2                               *
00110 #MSG,*****
00130 #MSG,*****
00140 #MSG,*                               NETWORK RUN AS SCHID OF 3                               *
00150 #MSG,*****
00170 #MSG,*****
00180 #MSG,*                               NETWORK RUN AS SCHID OF 4                               *
00190 #MSG,*****
I3 00 #JEND
00210 //SYSIN DD DUMMY
00220 //SYSPRINT DD SYSOUT=A
00230 //SYSUT2 DD DSN=CA-7.TEST1,
00240 // DISP=(NEW,CATLG,DELETE),UNIT=DISKA,VOL=SER=LIB112,SPACE=(TRK,1),
00250 // DCB=(RECFM=F,LRECL=80,BLKSIZE=80)
00260 //SYSUT1 DD *,DCB=BLKSIZE=80
00270 /*
00280 //STEP2 EXEC PGM=SASSTRLR,PARM=ACT
00290 //STEPLIB DD DSN=CA-7.LOADLIB,DISP=SHR
00300 //SYSPRINT DD SYSOUT=A
00310 //SYSOUT DD SYSOUT=A,DCB=BLKSIZE=133
-----+-----10-----+-----20-----+-----30-----+-----40-----+-----50-----+-----60-----+-----70*

```

The following screen reflects the result of the I subcommand to insert three blank lines after line 200. An I3 (I3space) was entered in the sequence field of line 200. The inserted lines were incremented by 10, the line increment value on the status line. The previous line 210 has been renumbered to 240. All subsequent lines are renumbered as necessary to retain ascending sequence numbers in the EWF.

Now a DD is placed in sequence lines 150 and 200. See FEM Sample Screen 7 on page 11-25 for the results.

FEM Sample Screen 6

```

-----+-----10-----+-----20-----+-----30-----+-----40-----+-----50-----+-----60-----+-----70*
PF                                          NOFILL I(010)
00100 #MSG,*                          NETWORK RUN AS SCHID OF 2                      *
00110 #MSG,*****
00130 #MSG,*****
00140 #MSG,*                          NETWORK RUN AS SCHID OF 3                      *
DD150 #MSG,*****
00170 #MSG,*****
00180 #MSG,*                          NETWORK RUN AS SCHID OF 4                      *
00190 #MSG,*****
DD200 #JEND
00210
00220
00230
00240 //SYSIN DD DUMMY
00250 //SYSPRINT DD SYSOUT=A
00260 //SYSUT2 DD DSN=CA-7.TEST1,
00270 // DISP=(NEW,CATLG,DELETE),UNIT=DISKA,VOL=SER=LIB112,SPACE=(TRK,1),
00280 // DCB=(RECFM=F,LRECL=80,BLKSIZE=80)
00290 //SYSUT1 DD *,DCB=BLKSIZE=80
00300 /*
00310 //STEP2 EXEC PGM=SASSTRLR,PARM=ACT
-----+-----10-----+-----20-----+-----30-----+-----40-----+-----50-----+-----60-----+-----70*

```


The following screen reflects the result of the DD subcommand to delete multiple lines from the EWF. A DD was placed in the sequence field of lines 150 and 200 of FEM Sample Screen 6 on page 11-24. Lines 150 through 200 are deleted from the EWF.

FEM Sample Screen 7

```

-----+-----10-----+-----20-----+-----30-----+-----40-----+-----50-----+-----60-----+-----70*
PF                                NOFILL I(010)
00100 #MSG,*                      NETWORK RUN AS SCHID OF 2                                *
00110 #MSG,*****
00130 #MSG,*****
00140 #MSG,*                      NETWORK RUN AS SCHID OF 3                                *
00210
00220
00230
00240 //SYSIN DD DUMMY
00250 //SYSPRINT DD SYSOUT=A
00260 //SYSUT2 DD DSN=CA-7.TEST1,
00270 // DISP=(NEW,CATLG,DELETE),UNIT=DISKA,VOL=SER=LIB112,SPACE=(TRK,1),
00280 // DCB=(RECFM=F,LRECL=80,BLKSIZE=80)
00290 //SYSUT1 DD *,DCB=BLKSIZE=80
00300 /*
00310 //STEP2 EXEC PGM=SASSTRLR,PARM=ACT
00320 //STEPLIB DD DSN=CA-7.LOADLIB,DISP=SHR
00330 //SYSPRINT DD SYSOUT=A
00340 //SYSOUT DD SYSOUT=A,DCB=BLKSIZE=133
00350 //SYSIN DD *,DCB=BLKSIZE=80
00360 /LOGON MASTER
-----+-----10-----+-----20-----+-----30-----+-----40-----+-----50-----+-----60-----+-----70*

```

The following screen reflects multiple subcommands entered on the screen.

FEM Sample Screen 8

```

-----+-----10-----+-----20-----+-----30-----+-----40-----+-----50-----+-----60-----+-----70*
PF                                NOFILL I(010)
00100 #MSG,*                      NETWORK RUN AS SCHID OF 2                                *
00110 #MSG,*****
00130 #MSG,*****
00140 #MSG,*                      NETWORK RUN AS SCHID OF 3                                *
00210
00220
00230
00240 //SYSIN DD DUMMY
00250 //SYSPRINT DD SYSOUT=A
P0260 //SYSUT2 DD DSN=CA-7.TEST1,
00270 // DISP=(NEW,CATLG,DELETE),UNIT=DISKA,VOL=SER=LIB112,SPACE=(TRK,1),
00280 // DCB=(RECFM=F,LRECL=80,BLKSIZE=80)
D0290 //SYSUT1 DD *,DCB=BLKSIZE=80
00300 /*
DD310 //STEP2 EXEC PGM=SASSTRLR,PARM=ACT
00320 //STEPLIB DD DSN=CA-7.LOADLIB,DISP=SHR
DD330 //SYSPRINT DD SYSOUT=A
00340 //SYSOUT DD SYSOUT=A,DCB=BLKSIZE=133
I2 50 //SYSIN DD *,DCB=BLKSIZE=80
00360 /LOGON MASTER
-----+-----10-----+-----20-----+-----30-----+-----40-----+-----50-----+-----60-----+-----70*

```

The following screen reflects the results of multiple subcommands entered on the screen displayed in FEM Sample Screen 8 on page 11-25. A P was placed in sequence number 260. A D was placed in sequence 290. A DD was entered in sequence numbers 310 and 330. An I2 and one blank were placed in sequence 350.

FEM Sample Screen 9

```

-----+-----10-----+-----20-----+-----30-----+-----40-----+-----50-----+-----60-----+-----70*
PF                                          NOFILL I(010)
00260 //SYSUT2 DD DSN=CA-7.TEST1,
00270 // DISP=(NEW,CATLG,DELETE),UNIT=DISKA,VOL=SER=LIB112,SPACE=(TRK,1),
00280 // DCB=(RECFM=F,LRECL=80,BLKSIZE=80)
00300 /*
00340 //SYSOUT DD SYSOUT=A,DCB=BLKSIZE=133
00350 //SYSIN DD *,DCB=BLKSIZE=80
00360
00370
00380 /LOGON MASTER
00390 DEMAND,JOB=DUSAXX02,LEADTM=100
00400 //STEP3 EXEC PGM=IEBGENER
00410 //SYSPRINT DD SYSOUT=A
00420 //SYSIN DD DUMMY
00430 //SYSUT2 DD DSN=CA-7.TEST2,
00440 // DISP=(NEW,CATLG,DELETE),UNIT=DISKA,VOL=SER=LIB112,SPACE=(TRK,1),
00450 // DCB=(RECFM=F,LRECL=80,BLKSIZE=80)
00460 //SYSUT1 DD *,DCB=BLKSIZE=80
00470 CONTROL CARD
00480 /*

-----+-----10-----+-----20-----+-----30-----+-----40-----+-----50-----+-----60-----+-----70*

```

11.5.2 Updating Text

The most common editing activity is to make changes to existing text. The following is a step-by-step example for this activity using the CA-7 text editor. In the example, the primary screen is the DB.7 screen, but the same steps apply to other screens, such as DB.4 and QM.5.

See Figure 11-1 on page 11-2.

To update text proceed as follows:

1. Display the primary screen, in this case, DB.7.
2. Type FE in the function field, then enter the required fields to identify the source of the text to be edited, such as MEMBER and DSN or JCL-ID, and press Enter.
3. Use the editor subcommands to make desired changes to the text.
4. Enter the SAVE or SR subcommand to retain the changes and return to the DB.7 screen. If SAVE is used to leave the editor, another function is required on the DB.7 screen to move the updated text to a permanent file. If SR is used to leave the editor, the source file identified in step 2 is updated. If the text updates are not to be saved, leave the editor using the EXIT subcommand.

11.5.3 Creating Text

Another common editing activity is to create text from scratch. The following is a step-by-step example for this activity using the CA-7 text editor. In the example, the primary screen is the DB.4 screen, but the same steps apply to other screens, such as DB.7 and QM.5.

See Figure 11-1 on page 11-2.

To create text, proceed as follows:

1. Display the primary screen, in this case, DB.4.
2. Since this is a menu screen, a secondary entry is required, such as 1 for job level documentation.
3. Once on the DB.4.1 screen, enter CLEAR in the function field to initialize the active area. (If the active size is 0, this step can be bypassed.)
4. Next enter EDIT in the function field to transfer to the editor. The edit screen returned indicates EWF EMPTY.
5. Enter I or INSERT which to invoke Data Insert processing.
6. Enter text as desired. Lines are double spaced with sequence numbers protected. If text is entered on all numbered lines displayed, another screen of blank lines is presented for continued entry. After all input is complete, leave at least one blank line on the screen and press Enter to return to FEM.
7. If changes are needed to the newly created text, use the editor subcommands to manipulate the data.
8. Enter the SAVE subcommand to retain the text and return to the DB.4.1 screen.
9. Type SAVE in the function field and the job name in the JOB field. Data may be entered in the other fields as well. Press Enter to add the text to the database. (If no data is to be entered in the other fields, SS can be done in Step 8 which does steps 8 and 9 with one command.)

11.5.4 Special Considerations

11.5.4.1 PF/PA Key Usage

While in the CA-7 text editor, PF and PA keys should not be used. These keys may be equated to top line commands not valid in the editor, and their use can cause unpredictable results. Even if the equated command works, all edit data is lost since the EWF is not saved to the Active Area.

11.5.4.2 Character Translation

CA-7 translates all character data to uppercase unless the text editor is used in Full Edit Mode. If using FEM, then character data will be translated or not depending on the case setting. If the case setting is 'UPPER', then all character data will be translated to uppercase. If the case setting is 'MIXED' then characters on data lines will be left untranslated. The FEM subcommands MIXED and UPPER may be used to change the case setting. The current case setting is reported on the status line.

On entry to the CA-7 editor in Full Edit Mode the case setting is 'UPPER' unless lower-case characters are detected. In that event the initial case setting is 'MIXED'.

If the case setting is changed to 'UPPER' then all character data in the editor will be translated to uppercase when it is saved to the active area. If data is not saved then the case setting only affects the current data display.

The case setting may not be changed unless INITCASE=Y is specified on the OPTIONS statement in the CA-7 Initialization File. If this value is not specified then the case setting is always set to 'UPPER' and cannot be changed.

When data is stored in the database or queue files, it is compressed using control codes for repeated characters.

11.5.4.3 Nondisplayable Data (hex)

Nondisplayable data contained in text edited using the CA-7 text editor can cause unpredictable screen output or a terminal disconnect. Avoid editing files containing such data.

11.6 TSO/ISPF Editor Environment

The CA-7 TSO/ISPF Interface allows for the fully integrated use of the ISPF editor where text editing is required. Some of the features described in 11.5, “CA-7 Text Editor Environment” on page 11-8 are available in the ISPF environment as well. There are however, important differences, and these are noted in this subtopic.

This subtopic explains the use of the ISPF editor only as supported under the CA-7 TSO/ISPF Interface.

In the following discussion, two types of CA-7 online terminal sessions are compared and contrasted:

- a *directly connected* CA-7 terminal session
- an *ISPF connected* CA-7 terminal session

A CA-7 online terminal session that is acquired outside of ISPF is considered *directly connected*. A CA-7 terminal session acquired from an ISPF session is considered an *ISPF connected* session.

Regardless of how the terminal session is connected, the file that is edited in CA-7 is always an EWF or Editor Work File. The format of the EWF differs depending on the type of terminal session where the editing occurs.

When editing takes place in a *directly connected* session (in a session where the TSO/ISPF interface is not being used), the EWF is a file internal to CA-7. Manipulation of data on the file is handled entirely by CA-7 in the CA-7 address space in response to CA-7 editor commands entered by the user.

In an *ISPF connected* session (in a CA-7 terminal session acquired under ISPF), the EWF is a temporary data set that is dynamically allocated and maintained by programs running under the ISPF session. All data manipulation takes place in the TSO user's address space using the ISPF editor.

In the TSO user's address space, the ISPF editor is used strictly for data manipulation. The primary source or target file (for example, a JCL library) is never updated by the ISPF session. All updates are handled by CA-7 from the CA-7 address space. The previous discussion detailing the flow of data between primary source/target, active area and EWF is valid not only for editing from a *directly connected* session but also from an *ISPF connected* session.

11.6.1 Requesting CA-7 Edit Functions from the ISPF Editor

There are several CA-7 functions which may be requested from the editor screens which cause termination of the edit session. These functions differ in the data movement that takes place when the function is requested. In a *directly connected* CA-7 terminal session, the subcommands used to request these functions are:

SAVE	Terminates the edit session and saves the contents of the EWF to the active area.
EXIT	Terminates the edit session without saving the contents of the EWF.
SS	Terminates the edit session, saves the contents of the EWF in the active area, and saves the contents of the active area to the primary source.
SR	Terminates the edit session, saves the contents of the EWF in the active area, and replaces the primary source with the contents of the active area.

SS and SR processing differs from the descriptions offered above if queue JCL is being edited.

All primary source/target updates are handled by CA-7; therefore, some means must be provided to allow the user to indicate to the interface programs from the editor, what CA-7 is to do with the data in the EWF when the edit session is terminated. In an *ISPF connected* CA-7 terminal session, the following ISPF edit macros may be used to request the CA-7 edit functions described above:

CA7SAVE	Terminates the edit session. Sends EWF to CA-7 and requests SAVE processing. Deletes the temporary data set (EWF).
CA7EXIT	Terminates the edit session. Sends EXIT request to CA-7. Deletes the temporary data set (EWF).
CA7SS	Terminates the edit session. Sends EWF to CA-7 and requests SS processing. Deletes the temporary data set (EWF).
CA7SR	Terminates the edit session. Sends EWF to CA-7 and requests SR processing. Deletes the temporary data set (EWF).

These are standard ISPF edit macros, provided as CLISTs. When one of these CLISTs is invoked, the value of an internal variable is set. The interface programs retrieve the value of the variable and request the appropriate CA-7 edit function (for example, SAVE, SS, SR, or EXIT).

11.6.2 Default Requests for CA-7 Edit Functions

In an *ISPF connected* CA-7 terminal session, a CA-7 editor function is always requested when the editor is exited. This function may be requested either explicitly, as in the case of one of the edit macros, or may be requested by default if the editor is exited in any other way (if, for example, PF03 is set to END and PF03 is pressed).

The CA-7 edit action that is requested by default is named in a message that appears when the editor is entered.

IF DATA IS SAVED, THEN A CA-7 xxxx WILL BE PERFORMED

Where:

xxxx may be either EXIT, SAVE, SS, or SR. The default action is set to SR initially. The default setting may be changed for subsequent edit sessions by issuing an edit macro. For example, if the default that appears in the message is SR, and if the CA7SS edit macro is used to terminate an edit session, then on the next session, the default setting that appears in the message is SS. This is the action that CA-7 performs if the edit session is terminated and data is saved. If data is not saved, CA7EXIT is requested by default. For example, suppose that the following message appears when the editor is entered for a member on a JCL library:

IF DATA IS SAVED, THEN A CA-7 SR WILL BE PERFORMED

Also suppose that an ISPF SAVE command is issued during the edit session. If the edit session is terminated without issuing a CA-7 edit macro then a CA-7 SR is requested. In this case, if the user did not want the data to replace the JCL member, a CA7EXIT would be required. An ISPF CANCEL would not suffice to prevent the SR from being issued, because an ISPF SAVE command was entered. The only way to avoid an implicit request for the default action is to enter the appropriate CA-7 edit macro. AUTOSAVE ON is a common edit profile setting and in many cases data may be saved automatically if data is changed. It is very important to be aware of the default CA-7 edit function that may be issued when leaving the edit session.

In the following example, text for an existing member of a JCL PDS is updated.

1. Request the JCL screen by typing DB.7 on the top line.
2. Enter FE in the field marked FUNCTION. Enter the member name in the field marked MEMBER. Enter all other information necessary to locate the member (JCLID or DSN and so forth), then press Enter.
3. The ISPF editor is invoked and data from the active area appears. Use ISPF editor commands to make desired changes.
4. Enter CA7SR from the command line to save the changes. To exit the editor without saving any changes, enter CA7EXIT from the command line. If the default CA-7 edit function is acceptable and if data was changed and if AUTOSAVE ON is set in the edit profile, then simply exit the editor using the ISPF END command (through a PF KEY if desired).

11.6.3 Special Considerations

11.6.3.1 ISPF Edit Profile Settings

In an *ISPF connected* CA-7 terminal session, the EWF is a *temporary* data set that is acquired only when needed by the interface programs and is deleted when the user exits the editor. It is a temporary data set and it is not retained in the event of abnormal termination, thus the recovery mode setting in the ISPF edit profile is of little benefit.

Although the ISPF editor command to set CAPS OFF is valid in an *ISPF connected* session, such a setting is effectively ignored since CA-7 is handling all output to the primary source/target. When the EWF is received, CA-7 translates all lowercase characters to uppercase prior to update of the active area.

Note: Edit facilities are restricted to character data whether the terminal session is *directly connected* or *ISPF connected*. Even though in an *ISPF connected* session HEX ON may be used to create unprintable hex data, this is strongly discouraged since CA-7's response to such data is unpredictable.

11.6.3.2 PF/PA Key Usage

Although both CA-7 and ISPF allow command input through a PF key, in the ISPF editor all PF keys are used for ISPF command input only. See the *CA-7 Systems Programmer Guide* for further information on PF keys in an *ISPF connected* CA-7 terminal session.

The use of PA keys is strictly subject to ISPF restrictions. PA key interrupts are not supported in an *ISPF connected* terminal session.

11.6.3.3 SUBMIT Function

If JCL is submitted using the ISPF SUBMIT function, CA-7 **does not track** the job. It is recommended that the SUBMIT function be deactivated through an ISPF command table for the CA7 application. See the *CA-7 Systems Programmer Guide* for further information on command table modifications for the CA7 application.

11.6.3.4 Size of Data

The maximum number of lines (or records) that can be edited using the ISPF interface is approximately 5000.

Chapter 12. Database Verification

Database verification consists of two basic database facilities, pointer verification and analyze commands. Pointer verification is used to verify contents of the database and executes in batch mode outside of CA-7. Analyze Commands are used to list, and optionally update, database contents from CA-7 terminals.

12.1 Pointer Verification

Module UCC7DBVR can be used to verify logical pointers in the database. When indicated by keywords from the control statements or PARM input, UCC7DBVR reads the database and checks pointers by record type.

12.1.1 Pointer Verification Parameters

The PARM or DBVPARM DD statement parameters available to request pointer verification are as follows:

IDSVERIFY

Verifies logical pointers for data sets, networks, and documentation members to the database Index Data Set (IDS) entries.

NWSCHD

Verifies that a one-to-one relationship exists between CA-7 networks having schedules and the schedules themselves.

JOBSCHD

Verifies that a one-to-one relationship exists between jobs having schedules and the schedules themselves.

ALLVSAM

Indicates the organization used for the CA-7 database. ALLVSAM indicates the database is totally VSAM as defined in the control statements residing in the DBPARMS data set. See the *CA-7 Systems Programmer Guide* for details on the contents of the DBPARMS data set. The same values used in the UCC7DBASE statements when the database was last loaded must be used here also to ensure correct access to the ALLVSAM database.

The parameters must appear in the execution JCL in either of the following formats:

```
PARM= 'ALLVSAM[,IDSVERIFY][,NWSCHD][,JOBSCHD]'
```

-or-

```
//DBVPARM DD *
ALLVSAM [,IDSVERIFY][,NWSCHD][,JOBSCHD]
```

When using the DBVPARM DD, the parameters must start in column 1.

JCL: The following is a sample of the pointer verification JCL:

```
//CA7DBVR JOB accounting info, etc.
//DBVR EXEC PGM=UCC7DBVR,PARM='dbvr keywords'
//STEPLIB DD DISP=SHR,DSN=CA-7.loadlib
//DBVOUT DD SYSOUT=A
//SYSUDUMP DD SYSOUT=A
//UCC7IDS DD DSN=user-defined-Index-data-set,DISP=SHR
//UCC7JLIB DD DSN=user-defined-Job-data-set,DISP=SHR
//UCC7DLIB DD DSN=user-defined-Dataset-data-set,DISP=SHR
//DBPARMS DD DSN=all-VSAM-data-base-parms,DISP=SHR (required for
/* ALLVSAM)
//DBVPARM DD * (or DD DUMMY if PARM used on EXEC statement)
database verification keywords go here or in PARM
/*
//
```

Note: A STEPCAT DD statement may be needed. If so, it follows immediately after the DBVR EXEC statement.

12.1.2 Pointer Verification by Record Type

The following shows how pointer verification proceeds by record type, the keyword which verifies each record type, and the corresponding error message numbers issued when an error is found. The error messages appear in numerical order in the *CA-7 Message Guide*.

12.1.2.1 Network Directory

IDSVERIFY keyword:

Checks Index database entries for correct pointer to network.

Issues error message 050.

NWSCHD keyword:

Checks pointer to network schedules.

Issues error messages 051 (input) and 052 (output).

12.1.2.2 Job Schedule Directory

JOBSCHD keyword:

Checks to see that the job schedule points to a job and that the job points back.

Issues error messages 053 and 054.

12.1.2.3 Data Set Directory

IDSVERIFY keyword:

Checks database index entry for triggering data set's (AUTO.DSnnnnnn) index pointer to the Dataset entry, and DSNBR correspondence between member and directory.

Issues error messages 059 through 062.

12.1.2.4 Documentation Directory

IDSVERIFY keyword:

Checks appropriate database index pointer for PP.network.NW, PP.jobname.JOB, and PP.system.SYS.

Issues error message 055.

12.1.2.5 Job Directory

IDSVERIFY keyword:

Checks Trigger Index database entries (JDEP.) for triggering jobs.

Issues error message 056.

JOBSCHD keyword:

Checks that job points to a schedule and that the schedule points back.

Issues error messages 057 and 058.

12.1.2.6 Data Set Member

IDSVERIFY keyword:

Checks database index entry for triggering data set's (AUTO.DSnnnnnn) index pointer to Dataset entry, and DSNBR correspondence between member and directory.

Issues error messages 059 through 062.

12.1.2.7 Network Member

IDSVERIFY keyword:

Checks index entry for network triggers (AUTO.NWnnnnnn) and index NW. entry pointing to network name.

Issues error messages 063 and 064.

12.1.2.8 Input Network Schedule Member

NWSCHD keyword:

Locates index network name and schedule member for network DSNBR, and checks that network points back.

Issues error messages 065 and 067.

12.1.2.9 Output Network Schedule Member

NWSCHD keyword:

Locates index network name and schedule member for network DSNBR, and checks that network points back.

Issues error messages 068 through 070.

12.1.2.10 Documentation Member

IDSVERIFY keyword:

Checks index pointers for PP.job.step.ddname, for existence of the database member and checks for correct index pointer for PP.data-set-name in the data set record.

Issues error messages 071 through 075.

12.1.2.11 Job Member

JOBSCHD,IDSVERIFY keywords:

Checks that the first record of a job member is a job entry.

Issues error message 011.

12.2 Analyze Commands

These commands are used to request an analysis and to list certain information from the CA-7 database and queues. Options are also provided which cause the database or queues to be updated during this analysis process. The Analyze commands and their respective database or queue elements are as follows:

PRRNJCL	Queue space for prior-run queue JCL
RESANL	Tape drives needed by the specified job(s)
RQMT	Predecessor job references
RQVER	Unsatisfied internal JOB/DSN requirements for request queue jobs
TRIG	Triggered jobs cross reference
XREF	Using-job references

Each command is discussed in detail in the *CA-7 Commands Guide*. These commands may be entered as top line commands through an online terminal or as a command through the Batch Terminal Interface facility. The presence of the command itself causes the list to be produced whether any updating is requested.

Many of the Analyze commands read large amounts of data and can tie up an online terminal for long periods of time. For this reason, it is recommended that the Batch Terminal Interface facility be used to issue these commands.

Analyze commands that update the database should only be used during periods of CA-7 low processing activity.

Additional database checking is performed by BACKUP, RELOAD, and database verification utilities. This chapter provides database verification utility information. See the *CA-7 Systems Programmer Guide* for information on the BACKUP and RELOAD utilities.

Chapter 13. Database Transportability

It may occasionally become necessary or desirable to transfer work from one CA-7 database to another. One reason for doing this might be to better balance the workload across multiple data centers. Another might be to move test application(s) into a production database from a test database. Certainly, there can be other reasons for transferring information between databases.

The effort required to manually redefine the workload into another database, even with online preformatted screens, can be sizeable. The amount of time typically available for this activity is limited. In most cases, a weekend or after-hours is selected; whenever production is relatively inactive. When the database move is done manually, the accuracy is suspect, particularly when it is done hurriedly.

CA-7 provides database transportability programs which, in batch mode and with the Batch Terminal Interface (BTI) facility, provide assistance in transfers of workload definitions.

The transportability process uses standard CA-7 inquiry, Batch Terminal Interface (BTI) and Database Maintenance (DBM) facilities to assist the user in relocating workload definitions from one CA-7 database to another or making mass changes. This process assumes that the workload to be moved or changed is in good running condition at the original (sending) site; that is, it is properly defined in the original site's database and presumably being run (or ready to be run) there on a regular basis.

Moving workload definitions from one database to another is a labor intensive process. The impact on both of the databases can be major. Certainly, the impact on any two (or more) data centers involved in such a move is major. The movement of:

- application data sets
- program libraries

and some other related items are beyond the realm of the functions performed by the database transportability process. This transportability process is only intended to simplify the CA-7 database portion of the total effort required. Some of the data sets and control reports generated by this process can be of direct assistance, however, in moving other items such as PROCLIBs.

The workload is placed in the new (receiving) database through various ADD commands of the standard DBM functions. Therefore, the database which is to receive the definitions of jobs and networks, can initially be empty or may contain definitions of other work. When other work already exists in the database, the incoming work is effectively merged into the database. The user should be sure that the incoming work does not create any conflicts with preexisting work. The incoming work should have unique names for:

- jobs
- networks
- data sets

and other items. Adding the work through DBM ensures that duplications do not occur.

This process does not actually add the work to the database. It simply creates data sets with which the user may perform necessary functions. Some of the data sets contain standard BTI commands for performing DBM functions. Other data sets are for handling CA-Librarian, CA-Panvalet, and PROCLIB members. (Movement of JCL can be suppressed if the user wishes to do that external to this process.)

These command data sets can be used anytime after the user has reviewed and/or altered the commands to accomplish the desired results. The data sets can then be used whenever and wherever they are needed. The data sets must be generated using the original site database, wherever it is located. Once created, the data sets can be sent to whatever location needs them. Copies of the control reports which correspond to the data sets could also be of value at the new site.

The user may find it helpful as a planning aid to run these jobs, multiple times if necessary, just to get the reports which are produced. These reports provide an excellent inventory organized into meaningful groups. That is, cataloged procedures are listed on one report, DBM work on other separate reports, and so forth.

The process could be repeated as many times as necessary to get the correct results without making any updates to any database. Once the user is satisfied that everything has been properly considered and provided for, the process can then be run to create the desired command data sets.

13.1 Assumptions

Not all of the process is automated. Some manual effort is required to ensure that the total problem is properly solved. There are also assumptions made which the user must consider when using this process. The following address some of the particular considerations for database movement using this process.

13.1.1 LOAD/RELOAD Status

All CPU jobs must have been LOADED with the current JCL to ensure that all data sets used by the job(s) are properly identified and handled in the move. Any associated data set documentation, cross-reference information, and so forth, are also dependent on this. Any job which was in RELOAD status at the original site may have some data set related errors in the new database definition. This is particularly true if JCL changes reflect data sets not previously used which also have documentation and special characteristics.

13.1.2 CA-11 Step Insertion

Whenever a job is defined in the original site database with INSERT-RMS=Y to cause the CA-11 step to be automatically inserted by CA-7, the job is moved assuming that CA-11 exists at the new site and the RMS step is to continue to be inserted automatically.

If that is not the case, the final commands must have the INSRTRMS=Y keyword manually edited (by the user) to the correct value before they are processed at the new site.

13.1.3 Base Calendars

This process does not move base calendars. The commands generated to define and resolve calendar schedules assume that the same calendars already exist at the new site. They are further assumed to have the same name(s) as those used at the original site.

The user must manually define the necessary calendars at the new site. If necessary, the BTI commands generated may be manually edited to reflect the correct calendar IDs before the BTI run is made.

13.1.4 Calendar Schedules

All calendar oriented schedules are assumed to have been RESOLVED against the proper Base Calendar. Without prior schedule resolution, the schedule is not correctly defined by this process. (Any differences in calendar names between the two sites must be manually edited into the BTI commands.)

13.1.5 DB.2.7 Schedule Modifications

This process has no facility to reapply any schedule changes which had been made through the DB.2.7 facility. Information on schedules, just as it would appear in LJOB or LSCHD output, is carried forward into the commands generated by this process. DB.2.7 changes have to be reapplied manually.

To assist the user in identifying those jobs which need such modifications reapplied at the new site, the SASSDT30-02 control report, produced in the last step, includes a warning message each time this condition is found. The message appears within a box of asterisks on the report following the RESOLV command for the job's schedules.

13.1.6 NXTCYC Settings

In the event that normal scheduling of CPU jobs has been altered by a NXTCYC command, that status is carried over to the database at the new site. This is accomplished by generating the appropriate NXTCYC command into the DBMADDS2 data set.

See the *CA-7 Commands Guide* for a discussion of the NXTCYC command.

13.1.7 Execution JCL Libraries

This process provides the commands and JCL statements necessary to add the JCL to a library at the new site. The execution JCL is added exactly as it appeared in LJCL output at the original site except for embedded blank records. Any records that are completely blank in positions 1-80 are ignored. (See 13.9, “Cataloged Procedures” on page 13-32 for how cataloged procedures are handled.) The commands use the same member names as those used at the original site.

It is also assumed that the new site uses the same number and types of JCL libraries as those used at the original site. This is typically the case. Using a different DSORG value at the new site makes it more difficult to use the generated commands since they are generated assuming that the same DSORG is used and the commands are quite different depending on the type of library being used.

It is further assumed that the libraries use the same JCLID or JCLLIB values. This is probably NOT a good assumption since the chance of duplication across two data centers is great, especially in the use of the default ID of zero. However, as long as a one-to-one relationship exists between the libraries at the two sites, all references to any one library in the generated BTI commands can be globally edited to another value to reflect whatever JCLID or JCLLIB value is used at the new site.

13.1.8 Cataloged Procedures

The user can bypass this feature by specifying JCL=N in the PARM for Job 2.

Otherwise, for each cataloged procedure executed in the production JCL, an IEHPROGM-type SCRATCH command is generated. These commands can optionally be used to delete the procedures at the original site once the move has been done and verified to be correct at the new site.

The list of these generated commands can assist the customer in identifying which cataloged procedures need to be moved to the PROCLIBs at the new site. Any such movement of data must be handled by the customer external to this process. The list is only an aid in accomplishing that task.

13.1.9 In-stream JCL Procedures

Although it is not a common practice for production jobs to use in-stream procedures (bound by PROC and PEND statements in the JCL), this process allows up to 256 such procedures within a single job. All other EXEC statements in the production JCL which execute procedures are assumed to be executing cataloged procedures and are handled as outlined in the prior discussion.

13.1.10 Workstation Networks

Only workstation networks directly related to a CPU job, through either a DB.3 or a DB.2.5 function, are automatically moved by this process. If there are other networks to be moved, they must be handled manually by the user as outlined in 13.4.4, “Unconnected Workstation Networks” on page 13-21.

13.1.11 User Level Documentation

Any documentation that was defined through the DB.4.3 screen is ignored by this process. If any such documentation is to be moved with the definition of the jobs, the customer must handle that manually, external to this process.

13.1.12 ANALYZE Application

To ensure database accuracy, the TRIG and RQMT functions of CA-7's ANALYZE application are performed by the first step of the process, prior to interrogating job dependencies and predecessor/successor relationships. The RQMT and TRIG commands are also included at the end of the generated DBM commands which perform deletes at the original site. This provides a much needed cleanup function following the deletion of the work being moved.

In the event of a rerun of this process, it is unnecessary to reexecute these commands at the original site. The processing time for each of these can be quite lengthy depending on the size of the database. In the event a rerun becomes necessary, simply delete those commands from the generated data set prior to executing the BTI run which uses them.

13.1.13 Commas in the Data

Since the information in the database is moved using BTI facilities, the use of commas in some of the data fields creates conflicts with the syntax requirements of batch commands. To help avoid such conflicts, this process replaces commas with blanks whenever they are found embedded within any of these data fields:

- User requirements (see DB.3.6)
- REPORT-ID value for documentation
- DESC value for documentation

If any such commas are required in the workload definition, they have to be manually reinserted in the new database after the data has been moved.

13.1.14 Sequence Numbers

All of the commands generated by this process are sequence numbered in positions 75-80. The characters U7 are also in positions 73-74 of the commands. All sequence numbers that previously existed within JCL statements are preserved by this process. That is, wherever a JCL member is embedded within the generated commands, the generated sequence is interrupted by the original JCL statements, all 80 positions. This should be considered when browsing the generated command data sets. Control reports listing the generated commands reflect the records, including sequence numbers, as they appear within the data sets. Each data set has its own range of sequence numbers.

13.1.15 User ID Security

All CPU jobs moved by this process carry the same User ID value to the new database. (See the UID field on the DB.1 screen on User ID on page 2-5 for more details on this facility.) If any of the jobs being moved create any conflicts with other User IDs in the new database, such conflicts must be resolved manually by the user before the generated commands are processed into the new database. New ID values could be manually edited into the generated commands any time before they are processed.

13.1.16 LINKed Documentation

Any documentation linked using the LINK field on any of the DB.4 screens cannot be linked by this process since the documentation number must be known at the time the add is done. It is assumed that the linked documentation is moved as some other piece of this process and any linked documentation which shows up during this process is ignored on that basis. The user must ensure that any linked documentation gets moved as part of some other job, system, and so forth. The linking of the documentation members then also has to be done manually after the adds have been made at the new site.

13.2 Database Extraction

This is an iterative process. It uses multiple BTI runs alternated with steps which examine database data. These steps produce command data sets which are used in another BTI run to explode the database definitions. The last step produces commands with which the user can accomplish the movement of the data from one database to another.

Standard LJOB and/or LSYS commands, are provided by the user to first identify those CPU jobs which are to be moved. These commands may use any combination of selection options desired but must omit the LIST= parameter since later programs expect the default formats. See the *CA-7 Commands Guide* for a discussion of those commands.

A BTI run is made using those commands. Output from those commands is examined by the program SASSDT10 which produces a control report titled "Jobs Requested to be Moved". The following is an example of this report:

SASSDT10-01 Jobs Requested to be Moved

SASSDT10-01				CA-7 DATABASE TRANSPORTABILITY						PAGE NO.		4	
DATE: mm/dd/yy				JOBS REQUESTED TO BE MOVED									
D													
U	JOB	----	JCL----	SYSTEM	USR	MAIN	PROSE	SCHED	--NUMBER OF--	LAST-RUN			
P	NAME	ID	MEMBER	-NAME-	-ID	-ID-	DSNBR	DSNBR	STP	DDS	RUNS	DATE/TIME	
	DUSAXX07	002	DUSAXX07	PAYROLL	164	ALL	000893	*NONE*	011	047	0062	yyddd/1732	
D	DUSAXX23	002	DUSAXX23	PAYROLL	164	SY3	001259	000384	005	021	0018	yyddd/1456	
	DUSAXX05	002	DUSAXX05	PAYROLL	164	ALL	*NONE*	*NONE*	005	036	0074	yyddd/1824	
NO. OF JOBS TO BE MOVED:					168								
D INDICATES 'DUPLICATE' WHICH WAS IGNORED													

Another data set is created containing more BTI commands necessary to review not only the job(s) but also the triggers, requirements, and JCL for those jobs. To accomplish this, SASSDT10 generates a series of three commands for each nonduplicate job name listed by the BTI step. Each of these commands is an LJOB command, with one each written for the options LIST=STEPDD, LIST=TRIG, and LIST=RQMT, in that sequence.

It is this examination of the triggers and requirements which makes it possible for the program to automatically include any workstation networks which are needed by these jobs. This also allows the program to look one level in each direction from the requested jobs to help ensure that all related jobs are being moved. Jobs overlooked in the initial LJOB and/or LSYS commands are thus included in the final output as long as they were no more than one level removed from the requested job.

After another BTI run, using the commands generated by SASSDT10, another program, SASSDT20, examines the output and creates yet another data set of BTI commands to list all of the database information that is needed to allow the work to be redefined into another database. These commands are then the input to another BTI run.

Another control report is produced that indicates which commands were generated in this step for each workload component to be moved. The commands generated by SASSDT20 are listed beneath the command generated by SASSDT10 which listed the component to be moved. If no commands were generated for any of the incoming commands, the characters ****NONE**** appear.

SSASSDT20-01 Expansion of Requested Jobs

SASSDT20-01	CA-7 DATABASE TRANSPORTABILITY	PAGE NO.	29
DATE: mm/dd/yy	EXPANSION OF REQUESTED JOBS		
JOB-NAME			
- COMMANDS GENERATED	-----	SEQ.-NO.	
DUSAXX05 (LIST=STEPDD)			
LPROS,SYS=PAYROLL		U7001722	
LJOB,JOB=DUSAXX05,LIST=ALL		U7001723	
LJCL,JOB=DUSAXX05		U7001724	
LPROS,JOB=DUSAXX05		U7001725	
LPROS,JOB=DUSAXX05,STEP=STEP0050,DD=HISTFILE		U7001726	
LDSN,DSNBR=DS008162,LIST=TRIG		U7001727	
LPROS,DSN=PAYROLL.HOURLY.HISTORY.DETAIL		U7001728	
LNTWK,NW=TIMECARD,LIST=ALL		U7001729	
LPROS,NW=TIMECARD		U7001730	
DUSAXX05 (LIST=TRIG)			
NONE			
DUSAXX05 (LIST=RQMT)			
LJOB,JOB=DUSAXX09,LIST=ALL		U7001731	
LJCL,JOB=DUSAXX09		U7001732	
LPROS,JOB=DUSAXX09		U7001733	

To assist the user with the separate task of getting all of the necessary data sets physically moved to the new site, a file of all data set names used by the jobs being moved is also produced by SASSDT20. The ddname of this file is DATASETS. This is a card-image data set containing the data set names beginning in position 1 of each record. All records written to this data set contain a data set name. (No heading lines.) The records are produced as the unique data set names occur in the data and therefore are not in any particular meaningful sequence. The file produced can be listed as is or could be sorted in data set name sequence prior to printing if desired.

Another BTI job is run using BTI commands produced by SASSDT20. When that run has completed, the output produced contains all of the information needed to generate commands with which the user can perform the move. The output from this BTI run is used as input to the last program in this process, SASSDT30. Several card-image data sets are created by SASSDT30.

Some of these are BTI data sets containing DBM commands, others are for handling JCL libraries and are in another format.

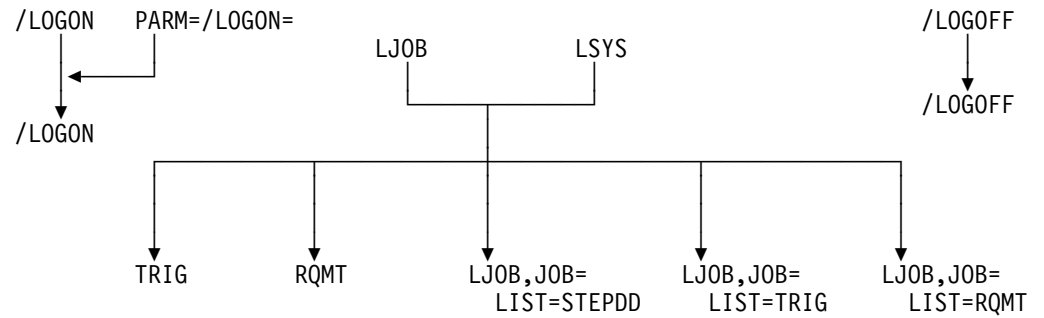
There are two data sets containing add type commands for the new CA-7 database. That is necessary since the processing sequence of the commands is not the same as the sequence in which the data becomes available to the programs which generate the commands. For example, a JOBCONN function must await the definition of both the connected items in the database before it can be performed. The user must process these two data sets in the correct sequence at the new site to ensure correct definition of the work in the database.

These data sets can be used whenever and wherever they are needed to accomplish the appropriate database activity. They probably require several changes which the user must make manually, using some text editor. Each data set created has a corresponding control report to allow the user to review the data before making any changes.

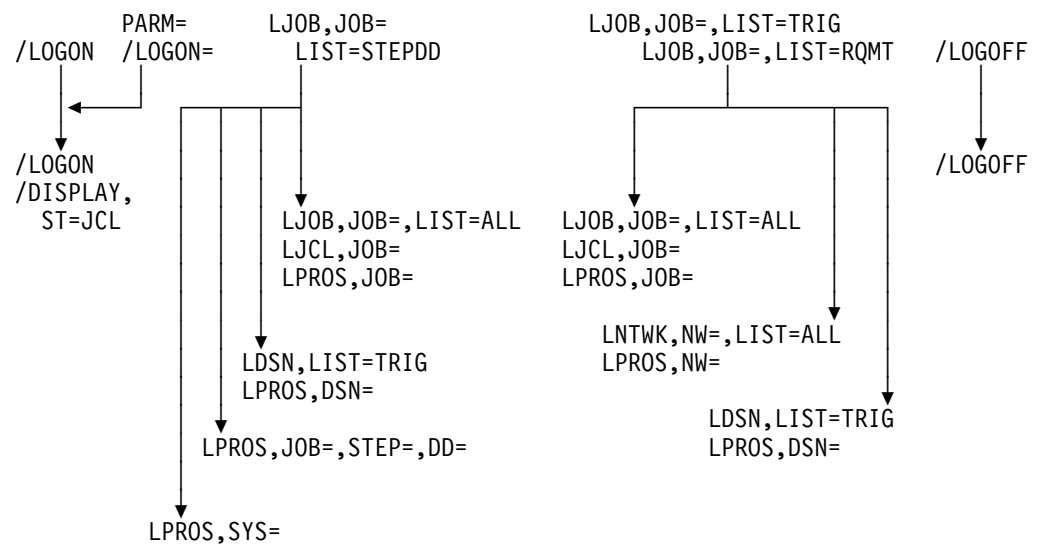
The following figure shows how the final commands produced evolve from the initial simple commands, with each one feeding subsequent commands. This sequence of commands, the same sequence that would be required to manually examine the workload for a move done without this process, could be repeated at an online CA-7 terminal to verify this process.

Command Evolution Process

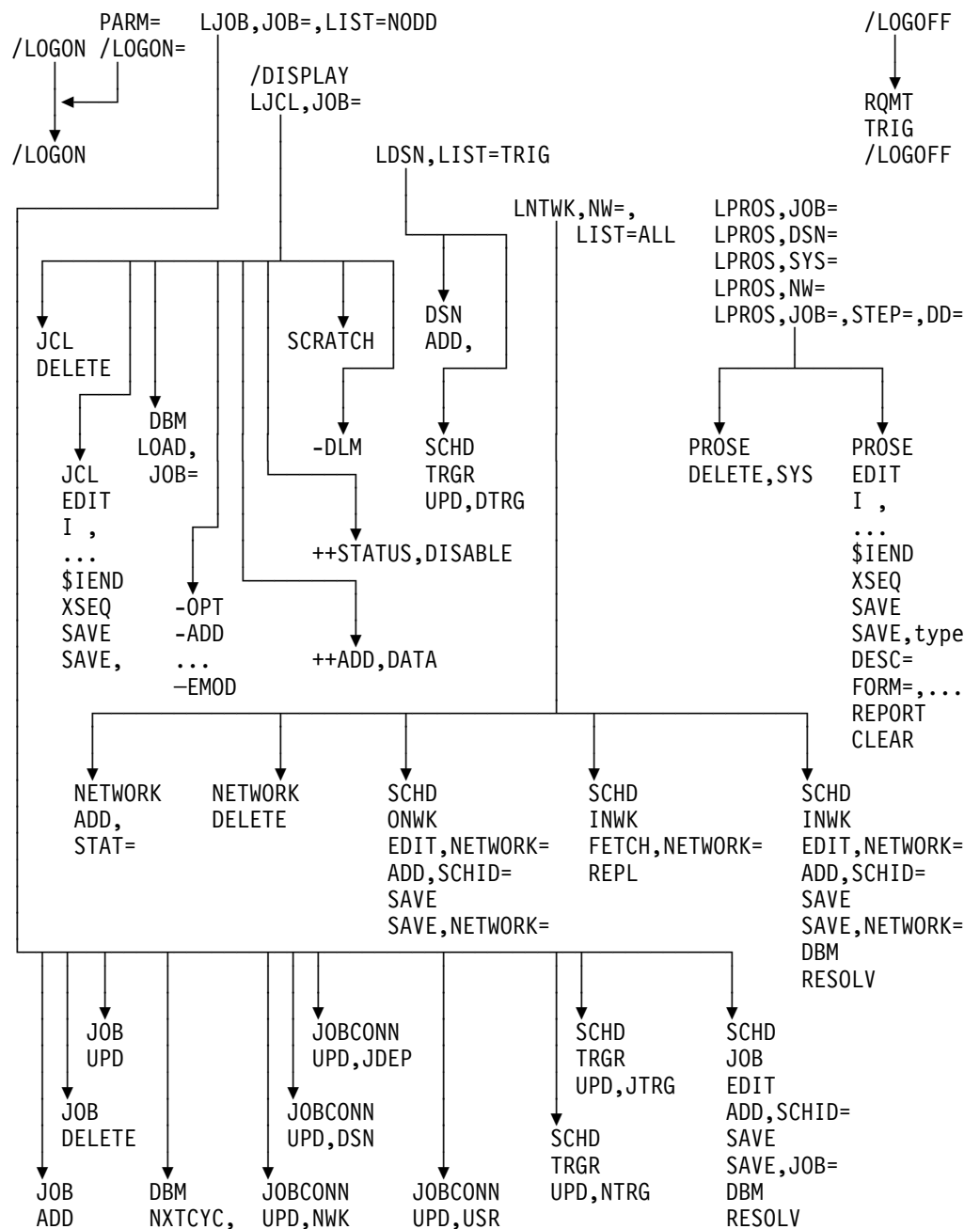
Module SASSDT10:



Module SASSDT20:



Module SASSDT30:



The following topics discuss the specific types of workload definitions handled by this process and describe what is produced to allow the workload definitions to be moved.

13.3 CPU Jobs

13.3.1 Adding at New Site

For each CPU job to be moved, a set of commands is generated into the DBMADDS1 data set to perform the necessary DBM JOB function. Those commands are generated as follows:

```
JOB
ADD,jobname,.....keywords and values.....,
.
.....more keywords and values.....,
.
```

See 2.1, “Adding a Job” on page 2-2 and 2.2, “DB.1 - CPU Job Definition Screen” on page 2-3 for further explanation of the content and requirements for these commands.

Some historical information such as number of times run and number of times late are not carried forward to the new site since there are no DB.1 screen input fields for those values. The average CPU and elapsed time requirements are carried forward.

There are a number of keywords which may be required to complete the definition of a job. Many of the available parameters are optional.

To minimize the number of parameters used, most parameters are not included in the generated commands if the value equals the usual CA-7 default. Multiple records may still be required to accommodate all of the needed keywords and values. Normal BTI continuation conventions are observed for those records.

The LOAD function needs to be performed for each of the jobs sometime after they are defined at the new site; therefore, the keyword RELOAD is included for all jobs with a value of Y (that is, RELOAD=Y). This allows the jobs to be defined at the new site without having to do a LOAD on each one of them at the time they are defined. This defers that process until the job runs for the first time at the new site. If the number of jobs is very high, it would take considerable time to perform individual LOADs all at once.

Since it is likely that the JCL library ID values are different at the new site, ADD commands for all CPU job definitions include the keyword JCLID, even if the default ID of zero was used. This makes it easier for the user to apply global changes to the commands if necessary.

These commands are sequence numbered in positions 73 through 80. They are also listed (with other DBM add functions for other related components being moved) on the

SASSDT30-01 Database Adds - Part 1 report. The following is an example of that report:

SSASSDT30-01 Database Adds - Part 1

SASSDT30-01	CA-7 DATABASE TRANSPORTABILITY	PAGE NO.	265
DATE: mm/dd/yy	DATABASE ADDS - PART 1		
- GENERATED COMMANDS ----- SEQ.-NO.			
JOB			U7017351
ADD,DUSAXX09,SYSTEM=PAYROLL,JOBNET=MONTHLY,USERID=164,JCLID=002,			U7017352
RELOAD=Y,RETJCL=Y,INSRTRMS=Y,CONDCODE=004,RELOPR=LT,PRTY=200			U7017353
JCL			U7017354
SAVE,DUSAXX09,JCLID=002			U7017355
NO. OF 'ADDS' FOR.			
JOB-	175	SYS PROSE-	2
JCL-	136	JOB PROSE-	174
NWKS-	248	NWK PROSE-	248
DSNS-	421	DD PROSE-	0
		DSN PROSE-	329

13.3.2 Requirements and Network Connections

Each CPU job being moved is examined to determine what predecessor and successor relationships exist in the database. These are determined by examining the connections that were defined in the database with the various DB.3 functions of CA-7. Those connection types and their corresponding batch DB.3 function are as follows:

```
JOB - JOBCONN,JOB
DSN - JOBCONN,DSN
NWK - JOBCONN,NWK
USR - JOBCONN,USR
```

Note: RPT connections are not included in this list and must be moved manually.

13.3.2.1 JOB Connections

For each of the JOB type connections, a set of commands is generated into the DBMADDS2 data set to perform the JOBCONN,JDEP function as follows:

```
JOBCONN
UPD,JDEP,jobname,OPT=A,SCHID=nnn,DEPJOB={-}depjobname [,LEADTM=nnnn]
                                     {/}
```

A negative dependency is indicated through the use of a / preceding the depjobname value. Prefixing the depjobname value with ? indicates a conditional dependency.

13.3.2.2 NWK Connections

For each of the NWK type connections, the following set of commands is generated into the DBMADDS2 data set to perform the JOBCONN,NWK function:

```
JOBCONN
UPD,NWK,jobname,OPT=A,NETWORK=networkname
      [,SCHID=nnn]      [,NWKSCHID=nnn]      [,LEADTM=nnnn]
      [,SUBID=xxxxxxx]  [,DESC=description]
```

13.3.2.3 DSN Connections

For each of the DSN type connections, the following set of commands is generated into the DBMADDS2 data set to perform the JOBCONN,DSN function:

```
JOBCONN
UPD,DSN,jobname,OPT=A,DSN=datasetname
    [,SCHID=nnn] [,LEADTM=nn] [,PERM=Y]
```

13.3.2.4 USR Connections

For each of the USR type connections, the following set of commands is generated into the DBMADDS2 data set to perform the JOBCONN,USR function:

```
JOBCONN
UPD,USR,jobname,OPT=A,USR=description [,SCHID=nnn]
```

To avoid conflicts with batch command syntax, any comma, parenthesis or apostrophe found within the defined description text is replaced with a blank.

All commands generated for JOBCONN functions are sequence numbered in positions 73 through 80. They are also listed on the SASSDT30-02 Database Adds - Part 2 report. SCHD commands for job, networks, and triggers are also produced in the DBMADDS2 data set and are listed on the report. The following is an example of that report:

SSASSDT30-02 Database Adds - Part 2

SASSDT30-02	CA-7 DATABASE TRANSPORTABILITY	PAGE NO.	119
DATE: mm/dd/yy	DATABASE ADDS - PART 2		
- GENERATED COMMANDS ----- SEQ.-NO.			
JOBCONN			U7008871
UPD,JDEP,DUSAXX09,OPT=C,SCHID=001,DEPJOB=DUSAXX10,LEADTM=99			U7008872
UPD,JDEP,DUSAXX09,OPT=C,SCHID=002,DEPJOB=/DUSAXX08			U7008873
SCHD			U7008874
TRGR			U7008875
UPD,JTRG,JOB=DUSAXX09,OPT=A,SCHID=001,TRGID=001,TJOB=DUSAXX11,			U7008876
DOTM=1730,LEADTM=0115			U7008877
NO. OF 'ADDS' FOR:		NXTCYC'S FOR:	
SCHD/JTRGS- 128	JOBCONN/JDEPS- 298	SKP- 2	
SCHD/NTRGS- 24	JOBCONN/NWKS- 248	OFF- 5	
SCHD/DTRGS- 4	JOBCONN/DSNS- 315		
JOB/SCHDS- 19	JOBCONN/USRS- 43		
NWK/SCHDS- 248			

13.3.3 Deleting at Original Site

For each job being moved, the following set of commands is generated into the DBMDELTS data set to perform the DBM JOB deletion function at the original site:

```
JOB
DELETE,jobname
```

The commands are sequence numbered in positions 73 through 80 and listed on the SASSDT30-03 Database Deletes report. See SASSDT30-03 Database Deletes Report on page 13-17 for a sample of that report. Some other deletion functions are also listed on that report since their commands are written to the DBMDELTS data set.

Deletion of the JCL for the job is performed separately with another set of commands. Deletion of associated documentation and schedules occurs automatically when the job is deleted at the original site.

The DB.1 screen function DD not only deletes a job but also definitions of any data sets for which this job was the last "using-job" reference. If the user wishes to use that function instead of the default DELETE function, the JOBDD keyword must be specified in the PARM data for Job 3 as discussed in 13.12.3, "Job 3" on page 13-46.

SASSDT30-03 Database Deletes

SASSDT30-03	CA-7 DATABASE TRANSPORTABILITY	PAGE NO.	18
DATE: mm/dd/yy	DATABASE DELETES		
*- GENERATED COMMANDS -----			SEQ.-NO.
JOB			U7001119
DELETE,DUSAXX09			U7001120
JCL			U7001121
DELETE,DUSAXX09,DSN=PAYROLL.JCLLIB,VOL=M80007			U7001122
NO. OF 'DELETES' FOR:			
	JOBS-	175	
	JCL-	136	
	NWKS-	248	
	SYS/PROSE-	2	

13.3.4 Disabling at Original Site

Job definitions at the original site should be retained in that database until it is confirmed that the job has run successfully at the new site at least once.

The DBMDELTS data set provides one set of commands to delete the definitions from the original database. The DBMSTOPS data set enables the user to disable the job at the original site without deleting the definition. This allows the definitions to be retained without the work being scheduled again automatically by CA-7 after the work has been moved.

These commands are generated as follows:

```
JOB
UPD,jobname,ADATE=yyddd,ATIME=hhmm
```

The values for ADATE and ATIME are, by default, taken from the system internal clock at the time that the command generation process is run. These values can be edited manually by the user to any other values prior to using them or the user can provide alternative values through the PARM keyword AFTER for Job 3 as discussed in 13.12.3, “Job 3” on page 13-46.

Sequence numbers are placed in positions 73 through 80 and the commands are listed on the SASST30-04 control report. The following is an example of that report:

(Any commands generated to disable workstation network schedules are also listed on that report since they are also generated into the DBMSTOPS data set.)

SASST30-04 Database Disables

```

SASST30-04          CA-7 DATABASE TRANSPORTABILITY      PAGE NO.   22
DATE: mm/dd/yy          DATABASE DISABLES

*- GENERATED COMMANDS -----* SEQ.-NO.

SCHD                      U7001098
INWK                      U7001099
FETCH,NETWORK=TIMECARD    U7001100
REPL                      U7001101
JOB                       U7001102
UPD,DUSAXX09,ADATE=00027,ATIME=1452  U7001103

NO. OF 'DISABLES' FOR:
      JOBS-      175
      SCHD/INWKS- 248

```

13.4 Workstation Networks

Workstation networks, both preprocessing and postprocessing, are handled by this process implicitly when a network is either connected to a CPU job being moved or triggers the execution of a CPU job being moved. (Only CPU jobs must be requested explicitly by the user for movement.)

If any workstation networks do not meet these criteria but must be moved, they must be moved manually by the user using the guidelines outlined in 13.4.4, “Unconnected Workstation Networks” on page 13-21.

13.4.1 Adding at New Site

For each network to be moved, commands to perform the DBM NETWORK function are generated into the DBMADDS1 data set as follows:

```
NETWORK
ADD,networkname,type,SUBID=subidvalue,JOB=jobname,
STAT=(stationname1,stationname2,...,stationname9)
```

See 8.1, “Adding a Network” on page 8-2 and 8.2, “DB.5 - Input/Output Network Definition Screen” on page 8-3 for further explanation of these commands. The values for these parameters are taken from information listed during an intermediate step of this process with a LNTWK command with the LIST=ALL option. See the *CA-7 Commands Guide* for a discussion of that command.

Each of these commands are sequence numbered in positions 73 through 80 and each are listed on the SASSDT30-01 control report. See SASSDT30-01 Database Adds - Part 1 Report on page 13-14 for an example of that report.

13.4.2 Deleting at Original Site

For each network name being moved, the following commands are generated into the DBMDELTS data set to assist in deleting the network definition from the original database:

```
NETWORK  
DELETE,networkname
```

The networkname is the same as it was in the commands generated to add the network at the new site.

Sequence numbers are placed in positions 73 through 80 of each command and each command is listed on the SASSDT30-03 control report. See SASSDT30-03 Database Deletes Report on page 13-17 for an example of that report.

13.4.3 Disabling at Original Site

13.4.3.1 Input Workstation Networks

There is no "don't schedule after" facility for workstation networks. The way to prevent networks from being scheduled again at the original site is to unRESOLVe the schedules for the networks in that database.

To accomplish this, one set of commands to perform the SCHD,INWK replace function is generated into the DBMSTOPS data set as follows:

```
SCHD  
INWK  
FETCH,NETWORK=networkname  
REPL
```

These commands are sequence numbered in positions 73 through 80 and are listed on the SASSDT30-04 control report. See SASSDT30-04 Database Disables Report on page 13-18 for an example of that report.

13.4.3.2 Output Workstation Networks

Since schedules for output workstation networks are based on the associated CPU job being scheduled and the CPU jobs are disabled through another set of commands, no other commands are required to disable these networks at the original site.

13.4.4 Unconnected Workstation Networks

To include any workstation networks which cannot be automatically included based on a connection or trigger definition, add two commands to the input (SYSIN DD) to 13.12.3, "Job 3" on page 13-46, as follows:

```
LNTWK,NW=networkname,LIST=ALL  
LPROS,NW=networkname
```

Include one set of commands for each unique network name to be included in the move. See Command Evolution Process on page 13-11 and the discussion 13.12, "Creating Command Files" on page 13-41 for more information on these facilities.

13.5 Application Data Sets

13.5.1 Adding at New Site

Since all of the jobs moved to the new site go through the CA-7 LOAD process before running at the new site, this process only generates commands to define data sets at the new site when any of the following is true:

- Data set is permanent to CA-7 globally
- Data set is permanent to individual specific job(s)
- Data set has documentation defined for it
- Data set triggers CPU job(s)

For each of these situations, commands to perform the DBM DSN function are generated into the DBMADDS1 data set as follows:

```
DSN
ADD,DSN=datasetname [,TYPE=PERM] ,SMF=Y
```

See 9.1, “Adding a Data Set” on page 9-2 and 9.2, “DB.6 - Data Set Definition Screen” on page 9-5 for further explanation of the requirements for these commands.

Adding data set definitions in this way reduces the overhead of each LOAD function when it is subsequently performed. It also ensures that the needed data set definitions are in the database so that the documentation can be added and/or any schedule triggering can be defined. Each data set is automatically assigned a new CA-7 data set number when the definition is made at the new site.

The TYPE=PERM parameter is only included for those data sets which were so designated in the old database.

The commands are sequence numbered in positions 73 through 80. They are also listed on the SASST30-01 control report. See SASST30-01 Database Adds - Part 1 Report on page 13-14 for an example of that report.

13.5.2 Deleting at Original Site

There are no commands generated to delete data set definitions at the original site. Effective with Version 2.8 of CA-7, the DB.1 screen has a function value of DD available. That function causes data set definitions to be deleted from the database at job deletion time if the job being deleted is the last "using-job" reference to the data sets.

If such a deletion is important:

- The JOBDD keyword can be used in the third job of the generation process, or
- The generated commands to delete jobs from the database at the original site can be manually edited to replace the function value DELETE with DD prior to using the generated commands.

13.12.2, "Job 2" on page 13-44 produces a file of data set names on the data set referenced by the ddname of DATASETS. This file contains one record for each unique data set name which is being moved by this process.

Such a file becomes a source from which batch commands could be produced to perform any desired deletions of data set definitions at the original site. A data set similar to any of the BTI data sets created by this process to do DBM functions could be produced and processed to clean up the original database once the definitions were no longer needed.

13.6 PDS JCL Members

If the user wishes to handle the movement of JCL totally external to this process, without any of these reports or commands being produced, the JCL=N parameter should be included in the PARM for module SASSDT20 as discussed later in 13.12, “Creating Command Files” on page 13-41.

The discussions in this chapter assume that the user wishes to produce reports and commands.

13.6.1 Adding at New Site

If a CPU job has an execution JCL member residing in a PDS library, a set of commands, including the JCL, are generated into the DBMADDS1 data set, to perform the DBM JCL function, as follows:

```
JCL
EDIT
I ,
.
.
. (JCL statements go here)
.
.
$IEND
XSEQ
SAVE
SAVE,membername,JCLID=nnn
CLEAR
```

The member name is the same as defined in the original database. The other parameters allow the JCL to be placed in a library with the same sequence numbers that existed before. See Chapter 10, “JCL Management” on page 10-1 and Chapter 11, “Edit Facility” on page 11-1 for further discussion of the commands used here.

JCL statements carry whatever sequence numbers previously existed in positions 73 through 80. The other statements are sequence numbered consecutively in positions 73 through 80.

Each JCL and SAVE,membername,JCLID=nnn statement produced are listed on the SASSDT30-01 control report. See SASSDT30-01 Database Adds - Part 1 Report on page 13-14 for an example of that report. The actual JCL statements and the other generated commands in each of these sets are not listed.

Only one such set of records is generated for any single member name no matter how many jobs may have used this member name for their JCL.

The JCL for jobs that already include a \$IEND cannot be moved with DBT. This JCL has to be copied externally from the DBT process.

13.6.2 Deleting at Original Site

Each JCL member copied into the DBMADDS1 data set for moving to a new site also has commands generated into the DBMDELTS data set to assist the user in cleaning up the JCL libraries at the original site. The commands are in the following format:

```
JCL  
DELETE,membername,DSN=datasetname,VOL=volser#
```

The membername is the member name used in the database. Values for DSN= and VOL= are taken from the definitions of the JCL libraries defined to CA-7.

These commands are sequence numbered in positions 73 through 80 in the DBMDELTS data set. Generated commands also are listed on the SASSDT30-03 control report. See SASSDT30-03 Database Deletes Report on page 13-17 for an example of that report.

Only one set of commands is generated for any single member name no matter how many jobs may have used this member name for their JCL.

13.7 CA-Librarian JCL

If the user wishes to handle the movement of JCL totally external to this process, without any of these reports or commands being produced, the JCL=N parameter should be included in the PARM for module SASSDT20 as discussed in 13.12, “Creating Command Files” on page 13-41.

The discussions in this chapter assume that the user wishes to produce reports and commands.

13.7.1 Adding at New Site

If a CPU job has an execution JCL member residing in a CA-Librarian library, a set of records, including the JCL, is placed in the LIBRADDs data set as follows:

```
-OPT NOINDEX,NORESEQ
-ADD membername,NOEXEC,NOLIST,NOPUNCH,SEQ=/73,8,10,10/
.
.
. (JCL statements go here)
.
.
-EMOD
```

The member name is the same as it was defined in the original database. See the CA-Librarian publications for the meanings of the various commands and parameters used here. -OPT, -ADD, and -EMOD statements are sequence numbered consecutively, in positions 73 through 80. JCL statements, with their own original sequence numbers are interspersed with these in the LIBRADDs data set. If the original sequence numbers are desired at the new site, the user must manually delete the SEQ parameter from the -ADD statement.

-OPT and -ADD statements produced are listed on the SASSDT30-06 control report which is produced on the file referenced by the ddname DT30CR06. The actual JCL statements and the -EMOD statements are not listed. See SASSDT30-06 CA-Librarian JCL Member Adds Report on page 13-27 for a sample of the SASSDT30-06 report.

Any other CA-Librarian control statements can be added to this data set prior to using it at the new site. Since CA-7 does not provide update access to CA-Librarian libraries, the user must assemble whatever job is needed to add this JCL to another CA-Librarian library at the new site. See the CA-Librarian publications to determine how this is done.

The new library at the new site and the JCLID or JCLLIB values in the JOB commands generated for the jobs which use this JCL must be coordinated to ensure correct definitions.

Only one such set of records is generated for any single member name no matter how many jobs may have used this member name for their JCL.

SASSDT30-06 CA-Librarian JCL Member Adds

SASSDT30-06	CA-7 DATABASE TRANSPORTABILITY	PAGE NO.	48
DATE: mm/dd/yy	CA-LIBRARIAN JCL MEMBER ADDS		
- GENERATED COMMANDS ----- SEQ.-NO.			
-OPT NOINDEX,NORESEQ			U7003673
-ADD DUSAXX03,NOEXEC,NOLIST,NOPUNCH,SEQ=/73,8,10,10/			U7003674
-OPT NOINDEX,NORESEQ			U7003676
-ADD DUSAXX07,NOEXEC,NOLIST,NOPUNCH,SEQ=/73,8,10,10/			U7003677
-OPT NOINDEX,NORESEQ			U7003679
-ADD DUSAXX05,NOEXEC,NOLIST,NOPUNCH,SEQ=/73,8,10,10/			U7003680
-OPT NOINDEX,NORESEQ			U7003682
-ADD DUSAXX09,NOEXEC,NOLIST,NOPUNCH,SEQ=/73,8,10,10/			U7003683
NO. OF MEMBERS ADDED- 1,228			

13.7.2 Deleting at Original Site

Each JCL member copied into the LIBRADDs data set for moving to a new site also has a command generated into the LIBRDELS data set to assist the user in cleaning up the JCL libraries at the original site. The command has the following format:

```
-DLM membername
```

The membername is the member name used in the database. See the CA-Librarian publications for other requirements.

Each -DLM command generated is listed on the SASSDT30-07 control report produced on the file referenced by the ddname DT30CR07. These commands are sequence numbered in positions 73 through 80 in the LIBRDELS data set.

Only one command is generated for any single member name no matter how many jobs may have used this member name for their JCL.

Following is a sample of the SASSDT30-07 report.

SASSDT30-07 CA-Librarian JCL Member Deletes

SASSDT30-07	CA-7 DATABASE TRANSPORTABILITY	PAGE NO. 24
DATE: mm/dd/yy	CA-LIBRARIAN JCL MEMBER DELETES	
- GENERATED COMMANDS ----- SEQ.-NO.		
-DLM DUSAXX01		U7001220
-DLM DUSAXX02		U7001221
-DLM DUSAXX04		U7001222
-DLM DUSAXX06		U7001223
-DLM DUSAXX08		U7001224
-DLM DUSAXX03		U7001225
-DLM DUSAXX07		U7001226
-DLM DUSAXX05		U7001227
-DLM DUSAXX09		U7001228
NO. OF MEMBERS DELETED- 1,228		

13.8 CA-Panvalet JCL

If the user wishes to handle the movement of JCL totally external to this process, without any of these reports or commands being produced, the JCL=N parameter should be included in the PARM for module SASSDT20 as discussed in 13.12, “Creating Command Files” on page 13-41.

The discussions in this chapter assume that the user wishes to produce reports and commands.

13.8.1 Adding at New Site

If a CPU job has an execution JCL member that resides in a CA-Panvalet library, a set of records, including the JCL, is placed in the PANVADDS data set as follows:

```
++ADD membername,DATA
.
.
.      (JCL statements go here)
.
.
```

The member name is the same as it was defined in the original database. The literal DATA preserves any sequence numbers being used in the JCL statements. ++ADD statements are sequence numbered consecutively, in positions 73 through 80. JCL statements, with their own original sequence numbers, are interspersed with these in the PANVADDS data set.

The ++ADD statements produced are listed on the SASSDT30-08 control report which is produced on the file referenced by the ddname of DT30CR08. The JCL statements themselves are not listed. See SASSDT30-08 CA-Panvalet JCL Member Adds Report on page 13-30 for a sample of the SASSDT30-08 report.

Any other CA-Panvalet control statements that the user may require can be added to this data set before using it at the new site. Since CA-7 does not provide update access to CA-Panvalet libraries, the user must assemble whatever job is needed to add this JCL to another CA-Panvalet library at the new site. See the CA-Panvalet publications to determine how this is done.

The new library at the new site and the JCLID values in the JOB commands generated for the jobs which use this JCL must be coordinated to ensure correct definitions at the new site.

Only one such set of records is generated for any single member name no matter how many jobs may have used this member name for their JCL.

SASSDT30-08 CA-Panvalet JCL Member Adds

SASSDT30-08	CA-7 DATABASE TRANSPORTABILITY	PAGE NO.	24
DATE: mm/dd/yy	CA-PANVALET JCL MEMBER ADDS		

- GENERATED COMMANDS -----	SEQ.-NO.
++ADD DUSAXX01,DATA	U7001220
++ADD DUSAXX02,DATA	U7001221
++ADD DUSAXX04,DATA	U7001222
++ADD DUSAXX06,DATA	U7001223
++ADD DUSAXX08,DATA	U7001224
++ADD DUSAXX03,DATA	U7001225
++ADD DUSAXX07,DATA	U7001226
++ADD DUSAXX05,DATA	U7001227
++ADD DUSAXX09,DATA	U7001228

NO. OF MEMBERS ADDED-	1,228
-----------------------	-------

13.8.2 Disabling at Original Site

Each JCL member copied into the PANVADDS data set for moving to a new site also has a command generated into the PANVDELS data set to assist in cleaning up the JCL libraries at the original site. The command has the following format:

```
++STATUS membername,DISABLE
```

The membername is the member name used in the database. See the CA-Panvalet publications for the meaning of the DISABLE parameter and any other requirements that exist for this purpose.

Each ++STATUS command generated is listed on the SASSDT30-09 control report produced on the file referenced by the ddname DT30CR09. These commands are sequence numbered in positions 73 through 80 in the PANVDELS data set.

Only one command is generated for any single member name no matter how many jobs may have used this member name for their JCL.

Following is a sample of the SASSDT30-09 report.

SASSDT30-09 CA-Panvalet JCL Member Deletes

SASSDT30-09	CA-7 DATABASE TRANSPORTABILITY	PAGE NO.	24
DATE: mm/dd/yy	CA-PANVALET JCL MEMBER DELETES		
- GENERATED COMMANDS ----- SEQ.-NO.			
++STATUS DUSAXX01,DISABLE		U7001220	
++STATUS DUSAXX02,DISABLE		U7001221	
++STATUS DUSAXX04,DISABLE		U7001222	
++STATUS DUSAXX06,DISABLE		U7001223	
++STATUS DUSAXX08,DISABLE		U7001224	
++STATUS DUSAXX03,DISABLE		U7001225	
++STATUS DUSAXX07,DISABLE		U7001226	
++STATUS DUSAXX05,DISABLE		U7001227	
++STATUS DUSAXX09,DISABLE		U7001228	
NO. OF MEMBERS DELETED- 1,228			

13.9 Cataloged Procedures

If the user wishes to handle the movement of JCL totally external to this process, without any of these reports or commands being produced, the JCL=N parameter should be included in the PARM for module SASSDT20 as discussed in 13.12, “Creating Command Files” on page 13-41.

The discussions in this chapter assume that the user wishes to produce reports and commands.

13.9.1 Adding at New Site

Normally, PROCLIBs are not made available to CA-7 since the processing of the production workload can be done without any such direct access. Therefore, this process makes no attempt to gain access to PROCLIBs. Movement of any cataloged procedures that may be required at the new site must be taken care of by the user by some other means.

See the following discussion of deleting PROCs at the original site for a hardcopy report that can be of assistance in identifying what cataloged procedures need to be moved. If nothing else, that report can be used as a checklist for the member names that need to be moved. If an entire library is being moved, this level of detail is not as important to the user.

13.9.2 Deleting at Original Site

Any PROCs executed by the JCL for any jobs being moved are candidates for deletion at the original site. Since the complete JCL is available to the SASSDT30 job, the JCL records are scanned to determine which cataloged procedures (PROCs) are being used. A table, accommodating up to 256 in-stream procedure names per job, enables the program to distinguish between in-stream and cataloged procedures.

If the cataloged procedures are no longer needed at the original site after the jobs are moved, the statements that are generated can be used to perform such optional deletions.

Only one statement is generated for each procedure name no matter how many jobs or steps execute it. These statements are generated into the DELPROCS data set and are listed on the SASSDT30-05 control report. See SASSDT30-05 PROCLIB Member Deletes Report on page 13-33 for a sample of that report.

Statements are generated in IEHPROGM format as follows:

```
SCRATCH MEMBER=membername,VOL=vvvvvv,DSNAME=
```

The member name is the same as it was in the JCL EXEC statements which referenced the PROC. The character string vvvvvv and space for a data set name value are provided in the statements. (CA-7 has no inquiry facility to determine those values for PROCLIBs.) A value to replace the character string vvvvvv and the proper value for the DSNAME= value must be manually provided by the user before using the generated statements.

SASSDT30-05 PROCLIB Member Deletes

SASSDT30-05	CA-7 DATABASE TRANSPORTABILITY	PAGE NO.	24
DATE: mm/dd/yy	PROCLIB MEMBER DELETES		
- GENERATED COMMANDS ----- SEQ.-NO.			
SCRATCH MEMBER=DUSAXX01,VOL=VVVVVV,DSNAME=		U7001220	
SCRATCH MEMBER=DUSAXX02,VOL=VVVVVV,DSNAME=		U7001221	
SCRATCH MEMBER=DUSAXX04,VOL=VVVVVV,DSNAME=		U7001222	
SCRATCH MEMBER=DUSAXX06,VOL=VVVVVV,DSNAME=		U7001223	
SCRATCH MEMBER=DUSAXX08,VOL=VVVVVV,DSNAME=		U7001224	
SCRATCH MEMBER=DUSAXX03,VOL=VVVVVV,DSNAME=		U7001225	
SCRATCH MEMBER=DUSAXX07,VOL=VVVVVV,DSNAME=		U7001226	
SCRATCH MEMBER=DUSAXX05,VOL=VVVVVV,DSNAME=		U7001227	
SCRATCH MEMBER=DUSAXX09,VOL=VVVVVV,DSNAME=		U7001228	
NO. OF PROCLIB MEMBERS- 1,228			

13.10 Documentation

13.10.1 Adding at New Site

CA-7 supports six categories of user-defined documentation. Five of these are handled by the transportability process. Any user level documentation (as defined through the DB.4.3 screen) is not handled. User documentation that has to be moved must be moved manually.

For each documentation member that is to be moved, a set of commands to perform the DBM PROSE function is generated into the DBMADDS1 data set in the following format:

```
PROSE
EDIT
I ,
.
. (documentation data goes here)
.
$IEND
XSEQ
SAVE
SAVE,type,membervalues,
DESC=description value ..... ,
FORM=formid,CARR=carriageid,TRAIN=trainid,COPIES=copyvalue,
REPORT=reportid
CLEAR
```

All commands generated are sequence numbered in positions 73 through 80.

The type and membervalues parameters in the second SAVE command are one of the following based on the type of documentation being moved:

Type	Parameters
System	SYS,SYSTEM=systemname
DD	DD,JOB=jobname,STEP=stepname,DD=ddname
Job	JOB,JOB=jobname
Network	NWK,NETWORK=networkname
Dataset	DSN,DSN=datasetname

See the DDPROSE keyword in 13.12.2, “Job 2” on page 13-44 for a user option for DD level documentation movement.

The value for a DESC keyword comes from any documentation description defined in the database. If no documentation description exists, the DESC keyword is not produced. To avoid any possible conflicts with batch command syntax, any commas found in the DESC text are replaced with spaces.

Values for the FORM, CARR, TRAIN, COPIES, and REPORT fields come from any corresponding values defined in the database. If no values were defined for these items, the command(s) are not produced. Any commas found within a defined report ID value are replaced with spaces.

Only the PROSE and SAVE,type,... commands are listed on the SASSDT30-01 control report. See SASSDT30-01 Database Adds - Part 1 Report on page 13-14 for an example of that report. The other commands are only written to the DBMADDS1 data set.

13.10.2 Deleting at Original Site

For SYS type documentation members ONLY, the following commands are generated into the DBMDELTS data set to delete that documentation at the original site:

```
PROSE  
DELETE,SYS,SYSTEM=systemname
```

All other documentation member deletions occur automatically whenever the job, data set, or network is deleted.

Only one set of these commands is generated for each system name. Each of these commands is sequence numbered in positions 73 through 80 and is listed on the SASSDT30-03 control report. See SASSDT30-03 Database Deletes Report on page 13-17 for an example of that report.

13.11 Schedules

13.11.1 Adding at New Site

13.11.1.1 CPU Job Schedules

For defined schedule IDs for those CPU jobs being moved, a set of commands to perform the necessary DBM SCHD,JOB function is generated into the DBMADDS2 data set as follows:

```
SCHD
JOB
EDIT
ADD,SCHID=nnn,ROLL=x,INDEX=>nnn,TIME=(dotm,ldtm,sbtm),
frequency values, etc. ...
.
.
SAVE
SAVE,JOB=jobname,SCAL=yy
DBM
RESOLV,SCAL=yy,TEST=NO,JOB=jobname
```

See Chapter 3, “Scheduling” for more information on these commands.

All of the values used here are as they were defined in the database including the Base Calendar ID. If the same Base Calendar names are not being used at the new site, these have to be manually edited by the user to the correct value.

Whenever there are multiple schedule IDs to be added for a single job, the ADD,... sequence is repeated as necessary, between the EDIT and SAVE commands, to define all those schedules at one time.

These commands are sequence numbered in positions 73 through 80 and listed on the SASSDT30-02 control report. See SSASSDT30-02 Database Adds - Part 2 Report on page 13-16 for an example of that report.

If the job had a current schedule modification, made with the online DB.2.7 screen function, a message to that effect is appended to the end of the listed records on report SASSDT30-02. Any such changes have to be investigated and reapplied manually at the new site after this schedule is defined in that database.

13.11.1.2 Job Triggers

For each job referenced as a triggered job by any of the jobs being moved, a set of commands to perform the DBM SCHD,JTRG function is generated into the DBMADDS2 data set as follows:

```
SCHD
TRGR
UPD,JTRG,JOB=jobname,OPT=A,SCHID=nnn,TJOB=triggeredjob,[,TRGID=nnn]
[,DOTM=hhmm] [,QTM=hhmm] [,LEADTM=hhmm] [,SBTM=hhmm]
```

See Chapter 3, “Scheduling” for more information on these commands.

The UPD command sequence is repeated as necessary, following the SCHD and TRGR commands, to accommodate all jobs triggered by the same job.

These commands are sequence numbered in positions 73 through 80 and are listed on the SASSDT30-02 control report. See SSASSDT30-02 Database Adds - Part 2 Report on page 13-16 for an example of that report.

13.11.1.3 Network Triggers

Whenever a CPU job being moved indicates that it is triggered by a workstation network, a set of commands to perform the SCHD,NTRG function is generated into the DBMADDS2 data set as follows:

```
SCHD
TRGR
UPD,NTRG,NWK=networkname,OPT=A,SCHID=nnn,TJOB=triggeredjob,
[,DOTM=hhmm] [,QTM=hhmm] [,LEADTM=hhmm] [,SBTM=hhmm]
```

See Chapter 3, “Scheduling” for more information on these commands.

The commands are sequence numbered in positions 73 through 80 and are listed on the SASSDT30-02 control report. See SSASSDT30-02 Database Adds - Part 2 Report on page 13-16 for an example of that report.

13.11.1.4 Data Set Triggers

For each CPU job shown as triggered by any data set accessed by any of the CPU jobs moved, a set of commands to perform the necessary SCHD,DTRG function is generated into the DBMADDS2 data set as follows:

```
SCHD
TRGR
UPD,DTRG,DSN=datasetname,OPT=A,SCHID=nnn,TJOB=triggeredjob [,TRGID=nnn]
[,DOTM=hhmm] [,QTM=hhmm] [,LEADTM=hhmm] [,SBTM=hhmm]
```

See Chapter 3, “Scheduling” for more information on these commands.

Since this process only moves jobs that are either explicitly requested or required by those jobs and within one level of those requested, this could produce commands that trigger some jobs that were not intended to be moved and may not have a job definition generated by this process. To prevent unsuccessful updates, the user has to verify the accuracy of this before trying to use the generated commands at the new site.

These commands are sequence numbered in positions 73 through 80 and are listed on the SASSDT30-02 control report. See SSASSDT30-02 Database Adds - Part 2 Report on page 13-16 for an example of that report.

13.11.1.5 Input Networks

For each input workstation network being moved to the new site, each schedule ID defined has a set of commands generated into the DBMADDS2 data set to perform the SCHD,INWK and RESOLV functions. These commands are in the following format:

```
SCHD
INWK
EDIT,NETWORK=networkname
ADD,SCHID=nnn,TIME=(dotm1,ldtm1,dody1,dotm2,...),
frequency values, etc. ...
.
.
SAVE
SAVE,NETWORK=networkname,SCAL=cc,ROLL=x,INDEX=>nnn
DBM
RESOLV,SCAL=cc,TEST=NO,NW=NW.networkname
```

See Chapter 3, “Scheduling” for more information on these commands. Also see 13.11.1.1, “CPU Job Schedules” on page 13-36 for a discussion of the DB.2.7 warning message which may appear on the control report following these records for workstation networks.

The ADD commands are repeated as many times as necessary between the EDIT and SAVE commands to accommodate all of the schedule IDs to be defined. All parameter values used are as they were defined in the original database including the Base Calendar ID which the schedule references. If the Base Calendar ID at the new site is different, these commands have to be manually edited to the correct value before being used.

Input network triggers are added. However, the TRGID value is not carried forward to the receiving site.

Each command is sequence numbered in positions 73 through 80 and is listed on the SASSDT30-02 control report. See SSASSDT30-02 Database Adds - Part 2 Report on page 13-16 for an example of that report.

13.11.1.6 Output Networks

For each output workstation network being moved, a set of commands to perform the necessary SCHD,ONWK function is generated into the DBMADDS2 data set as follows:

```
SCHD
ONWK
EDIT,NETWORK=networkname
ADD,SCHID=nnn,TIME=(dotm1,ldtm1,dody1,dotm2,...)
.
.
SAVE
SAVE,NETWORK=networkname
```

The ADD command may be repeated, between the EDIT and SAVE commands, as many times as necessary to define all of the schedule IDs that existed in the original database.

Each command is sequence numbered in positions 73 through 80 and is listed on the SASSDT30-02 control report. See SSASSDT30-02 Database Adds - Part 2 Report on page 13-16 for an example of that report.

13.11.2 Deleting at Original Site

13.11.2.1 CPU Job Schedules

Since the schedules for CPU jobs are automatically deleted whenever the job definition is deleted from the database, no additional commands are generated by this process to delete CPU job schedules.

See 13.3.4, “Disabling at Original Site” on page 13-18 for related commands to disable jobs.

13.11.2.2 Input Network Schedules

Since the schedules for input workstation networks are automatically deleted whenever the network definition is deleted from the database, no additional commands are needed to delete them.

13.11.2.3 Output Network Schedules

Since output networks are scheduled whenever their associated CPU job is scheduled, no extra commands are necessary. See 13.11.1.1, “CPU Job Schedules” on page 13-36 for more information on these schedules.

13.12 Creating Command Files

The transportability process consists of multiple steps. It could be done with a single job. The user probably wishes to develop multiple jobs, however, so that the processing can be manually verified between each of the three major iterations against the database.

The following discussions and examples assume that the user uses three different jobs to perform this process. JCL parameters which require special consideration are indicated in lowercase notation in the JCL examples. Unless unusually large numbers of jobs, quantities of documentation, and numbers of networks are being moved, the space allocations in the examples should be sufficient.

Before running any of the jobs, be sure to set the NLINE value in the TERM statement for the batch terminal being used to the maximum value. This minimizes the number of headings (and DASD space) required during the iterations of this process. See the *CA-7 Systems Programmer Guide* for a discussion of this initialization file parameter.

A discussion of the design limitations of this process is in 13.13, "Special Considerations" on page 13-50.

13.12.1 Job 1

The first two steps of this job are standard BTI steps. See "Using Batch Terminals" in the *CA-7 Interfaces Guide* for more information on these steps.

13.12.1.1 PARM Keywords

A CA-7 operator ID should be provided in the PARM data in the last step. A password can also be given. The format of the value for LOGON= is the same as it is for the /LOGON command to CA-7, including the comma which separates the operator ID and any password value being provided. Operator ID can be up to 8 characters in length. A maximum total of 65 characters is allowed for the operator ID, the password value and the comma which separates them. Any characters following the operator ID, and until a valid keyword (or end of PARM) is found, are assumed to be password data.

The LOGON keyword is optional. The default operator ID is MASTER in the generated /LOGON command and the command does not have a password value in it. If LOGON= is entered with no data following the equal sign, the /LOGON command is generated without an operator ID or a password.

If no LOGON value is provided or defaulted, the operator ID, and optionally the password field, has to be provided manually, in the DT10OUT data set, before that output data set can be processed by BTI (for SASSDT20).

13.12.1.2 Specifying Jobs to Move

The input to the BTI run is in the form of CA-7 LJOB and/or LSYS commands. They may be used in any combination to define the jobs to be moved. If there are any workstation networks to also be moved, the programs automatically include those with a predecessor or successor relationship with any of the jobs specified. If there are others not connected to a job, they must be moved manually by the user following the examples given in 13.4, “Workstation Networks” on page 13-19.

Requirements for /LOGON and /LOGOFF commands are the same here as for any BTI data set and must be supplied in addition to any LJOB or LSYS commands.

If the LSYS command is used, every job in the database with the same system name as that indicated has commands generated to define the job(s) into another database. This assumes that the optional system name was entered for each of the desired jobs when they were defined in the database. It may be safer to use the LJOB command with a generic job name since job names are required in the database. Job naming conventions usually make generic specification of the desired names relatively easy. Multiple commands can be used as necessary to define all of the desired jobs.

Jobs which have any type of predecessor or trigger relationship with any of the jobs being moved are also included in the final output command data sets produced by this process. That is made possible by the look-aside process which enables automatic inclusion of workstation networks. This look-aside process only looks one level in each direction from a job that was requested to be moved.

13.12.1.3 Data Sets Used/Created

The SYSPRINT output is being passed to the third step as that step's input.

The third step examines the output from the BTI step and generates another card-image BTI data set through ddname DT10OUT. This data set is the input to Job 2. The SASSDT10-01 control report, which can be of assistance in determining if this step ran correctly, is produced through the ddname DT10CR01. See SASSDT10-01 Jobs Requested to be Moved Report on page 13-8 for an example of that report.

```
//jobname JOB ..... ,REGION=1024K
//EXTRACT EXEC PGM=SASSBSTR,PARM=batch-terminal-id
//JOBLIB DD DSN=ca7.loadlib,DISP=SHR
//UCC7CMDS DD DSN=ca7.communications.data.set,DISP=SHR
//BATCHIN DD DSN=batch.input.data.set,DISP=SHR
//BATCHOUT DD DSN=batch.output.data.set,DISP=SHR
//SYSOUT DD SYSOUT=x
//SYSPRINT DD DSN=&DT10IN,DISP=(,PASS),UNIT=SYSDA,
//          SPACE=(CYL,(5,1),RLSE),DCB=(RECFM=FBA,LRECL=133,BLKSIZE=3990)
//SYSIN DD *
/LOGON operid[,password]
LJOB,JOB=jobname
LSYS,SYS=systemname
/LOGOFF
//*
//stepname EXEC PGM=SASSDT10,COND=(0,NE,EXTRACT),
//          PARM='LOGON=operid[,password]'
//DT10IN DD DSN=&DT10IN,DISP=(OLD,DELETE)
//DT10OUT DD DSN=input.to.job2,DISP=(,CATLG),
//          UNIT=SYSDA,SPACE=(CYL,(2,1),RLSE),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=4000)
//DT10CR01 DD SYSOUT=x,DCB=BLKSIZE=133
//SYSUDUMP DD SYSOUT=x
```

Figure 13-1. SASSDT10 Execution JCL

13.12.2 Job 2

After verifying that the jobs were identified correctly in Job 1, Job 2 may be run.

This job performs another iteration against the database much like the first job. Commands in the SYSIN data set for SASSBSTR cause listings to be produced for JCL, trigger, and requirement information for each of the jobs to be moved. Module SASSDT20 examines the BTI output (ddname DT20IN) and produces another card-image data set to be used in Job 3 (ddname DT20OUT).

13.12.2.1 PARM Keywords

A LOGON value should be provided in the PARM data for SASSDT20. This PARM value has the same format and meaning as discussed for SASSDT10 in 13.12.1, “Job 1” on page 13-41.

The user also has an option available which can significantly reduce the DASD requirements and slightly reduce the runtime. If the user does not use DD level documentation, as defined with the PROSE,DD online screen, DDPROSE=N should be specified in the PARM data. N is the only acceptable value.

Without this specification, the program produces an LPROS command for each nonblank stepname/ddname combination just to determine if any DD level documentation exists. Such unnecessary commands can represent as much as 85 percent or more of the number of commands needed to accomplish any one move. (Since it takes very little time for each of these commands to execute, the elapsed time savings are negligible compared to the DASD savings.) If the extra DASD space and runtime are of no particular concern, DDPROSE=N should not be specified. This ensures that any such existing documentation is included in the data being moved.

The user also has the option to totally ignore the movement of any JCL. This is accomplished by coding the parameter JCL=N in the PARM data for module SASSDT20. N is the only acceptable value. With this option specified, JCL is ignored here and the user must handle the movement of the JCL totally external to this process. JCL=N in this job suppresses all of the JCL related reports and command data sets in Job 3 as well. It may be wise to run these three jobs once without this option just to get the reports and again later with this option if necessary.

13.12.2.2 Data Sets Used/Created

DT20WORK is a work file used by SASSDT20. It must not have a DISP of MOD.

DATASETS contains one card-image record for each application data set name being moved with the name beginning in position 1 of the record. (This data set may be useful for producing sorted data set checklists to assist in the physical movement of data sets to the new site.)

The SASSDT20-01 control report (DT20CR01) can be used to verify accuracy before running Job 3. See SSASSDT20-01 Expansion of Requested Jobs Report on page 13-9 for an example of that report.

After verifying that Job 2 ran correctly, Job 3 may be run.

```
//jobname JOB ..... ,REGION=4M
//EXTRACT EXEC PGM=SASSBSTR,PARM=batch-terminal-id
//JOB LIB DD DSN=ca7.loadlib,DISP=SHR
//UCC7CMDS DD DSN=ca7.communications.data.set,DISP=SHR
//BATCHIN DD DSN=batch.input.data.set,DISP=SHR
//BATCHOUT DD DSN=batch.output.data.set,DISP=SHR
//SYSOUT DD SYSOUT=x
//SYSPRINT DD DSN=&DT20IN,DISP=(,PASS),UNIT=SYSDA,
//          SPACE=(CYL,(5,1),RLSE),DCB=(RECFM=FBA,LRECL=133,BLKSIZE=3990)
//SYSIN DD DSN=input.to.job2,DISP=(OLD,DELETE)
//*
//stepname EXEC PGM=SASSDT20,COND=(0,NE,EXTRACT),
//          PARM='LOGON=operid[,password][,DDPROSE=N][,JCL=N]'
//DT20IN DD DSN=&DT20IN,DISP=(OLD,DELETE)
//DT20OUT DD DSN=input.to.job3,DISP=(,CATLG),
//          UNIT=SYSDA,SPACE=(CYL,(2,1),RLSE),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=4000)
//DT20CR01 DD SYSOUT=x,DCB=BLKSIZE=133
//DATASETS DD SYSOUT=x,DCB=BLKSIZE=80
//DT20WORK DD DISP=(,DELETE),UNIT=SYSDA,
//          SPACE=(CYL,(1,1)),DCB=(RECFM=FBA,LRECL=133,BLKSIZE=3990)
//SYSUDUMP DD SYSOUT=x
```

Figure 13-2. SASSDT20 Execution JCL

13.12.3 Job 3

This job performs another iteration against the database, much like the first two jobs. Commands in the SYSIN data set, from Job 2, cause listings to be produced for other database elements that are to be moved. The user may also have manually included some LNTWK and LPROS commands to include some workstation networks that did not have an explicit connection to any of the jobs being moved. SASSDT30 examines the BTI output (DBMLISTS) and creates several command data sets.

13.12.3.1 PARM Keywords

A LOGON value should be provided in the PARM data for SASSDT30. This keyword value has the same format and meaning as that discussed for SASSDT10 in 13.12.1, “Job 1” on page 13-41.

The user also has the following optional keywords which can be provided in the PARM for the purposes indicated:

JOBDD=Y

Indicates to generate JOB deletes with the function value of DD instead of DELETE so that dormant data sets can be deleted when the job is deleted. DD commands require more execution time than DELETE commands whenever they are processed against the database to do the actual deletes. This time could be significant if a large number of commands are processed at once. See the discussion of the 2.2, “DB.1 - CPU Job Definition Screen” on page 2-3 for details of this function. Y is the only acceptable value.

AFTER=yyddd[hhmm]

Indicates "don't schedule after" values to use for the disable commands generated to disable CPU jobs at the original site. If not provided, the current system date and time are used for this purpose. If only the Julian date is provided, the default time-of-day is zeros (for the morning of the date entered). See the discussion of the 2.2, “DB.1 - CPU Job Definition Screen” on page 2-3 for details of this function.

BEFORE=yyddd[hhmm]

Indicates "don't schedule before" values to use for the add commands generated to add CPU jobs at the new site. If not provided, no value is generated and the work is eligible to run at the new site as soon as it is completely defined in that database. If only the Julian date is provided, the default time-of-day is zeros (for the morning of the date entered). See the discussion of the 2.2, “DB.1 - CPU Job Definition Screen” on page 2-3 for details of this function.

LOADS=Y

Indicates that LOAD commands are to be generated for each CPU job. When specified, LOAD commands are generated into the DBMADDS2 data set after the JCL has been processed for the job. This parameter cannot be used if JCL=N was specified in Job 2 since that specification prevents any JCL from appearing in the input to this job. (If necessary, multiple executions of this process could be run to create the desired command data sets.) Y is the only acceptable value.

NODSNS

Indicates that DSN ADD commands will not be generated in the output of this run. This parameter may be used if data set definitions are not required at the receiving site. By default, data set ADDs are generated.

13.12.3.2 Command Data Sets

Module SASSDT30 examines the BTI output (through ddname DBMLISTS) and produces card-image data sets containing the commands needed to accomplish the move. One control report is produced for each of the command data sets. The following are the data sets being created:

DDname	Commands	Report	Example
DBMADDS1	DBM add commands - part 1	DT30CR01	on SASSDT30-01 Database Adds - Part 1 Report on page 13-14
DBMADDS2	DBM add commands - part 2	DT30CR02	on SSASSDT30-02 Database Adds - Part 2 Report on page 13-16
DBMDELTS	DBM delete commands	DT30CR03	on SASSDT30-03 Database Deletes Report on page 13-17
DBMSTOPS	DBM disable commands	DT30CR04	on SASSDT30-04 Database Disables Report on page 13-18
DELPROCS	SCRATCH commands for PROCs	DT30CR05	on SASSDT30-05 PROCLIB Member Deletes Report on page 13-33
LIBRADDS	CA-Librarian JCL add modules	DT30CR06	on SASSDT30-06 CA-Librarian JCL Member Adds Report on page 13-27

DDname	Commands	Report	Example
LIBRDELS	CA-Librarian delete commands	DT30CR07	on SASSDT30-07 CA-Librarian JCL Member Deletes Report on page 13-28
PANVADDS	CA-Panvalet JCL add modules	DT30CR08	on SASSDT30-08 CA-Panvalet JCL Member Adds Report on page 13-30
PANVDELS	CA-Panvalet disable commands	DT30CR09	on SASSDT30-09 CA-Panvalet JCL Member Deletes Report on page 13-31

Details on the contents of these data sets and the examples of these control reports were given in the discussions on 13.3, “CPU Jobs” on page 13-13, and 13.4, “Workstation Networks” on page 13-19, and so forth.

It is very likely that some of these data sets require some manual editing and changing before they can be used. For example, changing JCLID numbers must be taken care of by editing these data sets. For other considerations, see 13.1, “Assumptions” on page 13-3 and 13.13.3, “Running at New Site” on page 13-51. The control reports closely mirror the contents of the data sets, including the generated sequence numbers. This makes it easier to locate specific commands requiring changes.

The card-image data sets must be preserved as necessary for subsequent use at whatever site is to process the commands.

DT30WORK is a work file used by SASSDT30. It must not have a DISP of MOD (the file is opened and closed multiple times).

If no CA-Panvalet or CA-Librarian JCL is being used, those data sets and their control reports may be DUMMYed or defined as NULLFILES.

```
//jobname JOB .....REGION=4096K
//EXTRACT EXEC PGM=SASSBSTR,REGION=1024K,PARM=batch-terminal-id
//JOBLIB DD DSN=ca7.loadlib,DISP=SHR
//UCC7CMDS DD DSN=ca7.communications.data.set,DISP=SHR
//BATCHIN DD DSN=batch.input.data.set,DISP=SHR
//BATCHOUT DD DSN=batch.output.data.set,DISP=SHR
//SYSPRINT DD DSN=&DT30IN,DISP=(,PASS),UNIT=SYSDA,
//          SPACE=(CYL,(8,1),RLSE),DCB=(RECFM=FBA,LRECL=133,BLKSIZE=3990)
//SYSIN DD DSN=input.to.job3,DISP=(OLD,DELETE)
//*
//stepname EXEC PGM=SASSDT30,COND=(0,NE,EXTRACT),
//          PARM='LOGON=operid[,password].....'
//SYSUDUMP DD SYSOUT=x
//DBMLISTS DD DSN=&DT30IN,DISP=(OLD,DELETE)
//DBMADD1 DD DSN=batch.DBM.add.commands.part.1,DISP=(,CATLG),UNIT=SYSDA,
//          SPACE=(CYL,(5,1),RLSE),DCB=(RECFM=FB,LRECL=80,BLKSIZE=4000)
//DBMADD2 DD DSN=batch.DBM.add.commands.part.2,DISP=(,CATLG),UNIT=SYSDA,
//          SPACE=(CYL,(5,1),RLSE),DCB=(RECFM=FB,LRECL=80,BLKSIZE=4000)
//DBMDELTS DD DSN=batch.DBM.delete.commands,DISP=(,CATLG),UNIT=SYSDA,
//          SPACE=(CYL,(1,1),RLSE),DCB=(RECFM=FB,LRECL=80,BLKSIZE=4000)
//DBMSTOPS DD DSN=batch.DBM.disable.commands,DISP=(,CATLG),UNIT=SYSDA,
//          SPACE=(CYL,(2,1),RLSE),DCB=(RECFM=FB,LRECL=80,BLKSIZE=4000)
//DELPROCS DD DSN=scratch.commands.for.procs,DISP=(,CATLG),UNIT=SYSDA,
//          SPACE=(CYL,(1,1),RLSE),DCB=(RECFM=FB,LRECL=80,BLKSIZE=4000)
//LIBRADD1 DD DSN=librarian.add.commands,DISP=(,CATLG),UNIT=SYSDA,
//          SPACE=(CYL,(1,1),RLSE),DCB=(RECFM=FB,LRECL=80,BLKSIZE=4000)
//LIBRDEL1 DD DSN=librarian.delete.commands,DISP=(,CATLG),UNIT=SYSDA,
//          SPACE=(CYL,(1,1),RLSE),DCB=(RECFM=FB,LRECL=80,BLKSIZE=4000)
//PANVADD1 DD DSN=panvalet.add.commands,DISP=(,CATLG),UNIT=SYSDA,
//          SPACE=(CYL,(1,1),RLSE),DCB=(RECFM=FB,LRECL=80,BLKSIZE=4000)
//PANVDEL1 DD DSN=panvalet.disable.commands,DISP=(,CATLG),UNIT=SYSDA,
//          SPACE=(CYL,(1,1),RLSE),DCB=(RECFM=FB,LRECL=80,BLKSIZE=4000)
//DT30WORK DD DISP=(,DELETE),UNIT=SYSDA,
//          SPACE=(CYL,(1,1)),DCB=(RECFM=FBA,LRECL=133,BLKSIZE=3990)
//DT30CR01 DD SYSOUT=x,DCB=BLKSIZE=133
//DT30CR02 DD SYSOUT=x,DCB=BLKSIZE=133
//DT30CR03 DD SYSOUT=x,DCB=BLKSIZE=133
//DT30CR04 DD SYSOUT=x,DCB=BLKSIZE=133
//DT30CR05 DD SYSOUT=x,DCB=BLKSIZE=133
//DT30CR06 DD SYSOUT=x,DCB=BLKSIZE=133
//DT30CR07 DD SYSOUT=x,DCB=BLKSIZE=133
//DT30CR08 DD SYSOUT=x,DCB=BLKSIZE=133
//DT30CR09 DD SYSOUT=x,DCB=BLKSIZE=133
```

Figure 13-3. SASSDT30 Execution JCL

13.13 Special Considerations

13.13.1 Design Limitations

The transportability process uses internal tables to keep from duplicating definition of jobs, networks, and other items in the final output. These tables also ensure that a minimum of commands are issued during the intermediate steps of this process.

Each of the three modules executed in the three jobs which constitute this process uses tables for these purposes as follows:

Module	Maximum#	Component
SASSDT10	50,000	CPU job names
SASSDT20	1,000	Application system names
SASSDT20	50,000	CPU job names
SASSDT20	10,000	Workstation network names
SASSDT20	300,000	CA-7 assigned data set numbers
SASSDT30	50,000	Execution JCL member names
SASSDT30	50,000	Cataloged procedure names
SASSDT30	256	In-stream procedures (maximum per job)

All but one of these limits apply to the entire workload definition being moved. The only exception is the limit of in-stream procedures, in SASSDT30, which is the maximum number within any one execution JCL member. (This table is used to distinguish between cataloged procedures and in-stream procedures.)

If any of these limitations are insufficient for any particular move, the user could:

- Break it up into multiples of this process by only requesting a number of jobs that does not exceed these limits, or
- Make the coding changes to the programs for the appropriate values.

13.13.2 Adding to the New Database

Once the appropriate command data sets have been physically taken to the new site, the DBM adds (parts 1 and 2) must be processed by CA-7 at that site. Any JCL libraries for CA-Panvalet or CA-Librarian JCL also must be processed at that site. Either a batch terminal job or a batch execution of CA-7 could be used to accomplish this. If using the batch execution of CA-7, see the *CA-7 Systems Programmer Guide* Chapter 3, for batch execution procedures.

Note: The DBMADDS1 and DBMADDS2 data sets must be processed in the correct sequence. If DBMADDS1 is not processed first, all of the commands in DBMADDS2 are rejected with error messages.

The user should review the amount of available space in the database at the new site before attempting to add the new jobs. Comparing job counts and space allocations at both sites, prior to performing this process, gives the user some idea of how much more space, if any, is needed. This probably was already done when the initial plans were being made for the movement of the workload to the other center. See the *CA-7 Systems Programmer Guide* for guidelines for calculating space requirements in the database.

Depending on the number of jobs, documentation lines, and other items being moved, the volume of output from DBM when the commands are processed at the new site could be very large. Normal batch terminal allocations would probably not be large enough to hold all of this. Since documentation is added one record at a time, with a response back after each line in batch, this alone can require a large allocation.

The output from the maintenance against the new database should be carefully reviewed for accuracy of definitions. Duplicates become obvious at this point. Any duplication of names for jobs, networks, and other items have to be resolved manually by the user.

13.13.3 Running at New Site

The work is eligible to be scheduled at the new site as soon as the work and its schedules are completely defined in the database. If any delay is desired, the commands for the CPU jobs can include the "don't schedule before" parameters, BDATE and BTIME, to accomplish this for CPU jobs (not networks).

The user can provide this either by:

- Using the BEFORE keyword in the PARM for Job 3, or
- Manually editing the desired values into the generated records prior to adding the work to the database.

13.13.4 Mass Changes at Existing Site

Even though the database transportability programs were developed to assist with the transfer of work from one site to another, it can also be used to perform mass or global changes at an existing site.

To accomplish such changes the desired work is identified by LJOB and LSYS commands, as described in preceding chapters, and the three jobs are run as indicated. Then desired changes are made to the command files (DBMADDS1 and DBMADDS2). Such changes could be renaming of jobs, systems, and data sets or just changing certain job options. Next the ADD functions would be changed to UPD if not changing job or data set names and any unnecessary commands deleted. Finally, a BTI job would process the command files to apply the changes to the existing database.

After all changes are verified, use the DBMSTOPS or DBMDELTS to remove any obsolete references.

13.14 Virtual Resource Management Database Extracts

The Virtual Resource Management (VRM) Database Transportability facility allows the selection of jobs with connected resources from the VRM database component and generates a set of CA-7 Batch Terminal Interface commands, in a fixed block file format, for transport to a different database.

The SASSDT60 program is used to read the VRM database component and select jobs based on user-specified criteria. The program creates a DT60ADDS file which contains the CA-7 Batch Terminal Interface commands required to add the job to resource connections to a different VRM database component. In addition to the CA-7 Batch Terminal Interface command file, a job selection report SASSDT60-01 is generated to report on any or all jobs selected for this run.

SASSDT60-01 DATE: 10/22/00		CA-7 DATABASE TRANSPORTABILITY VRM JOB EXTRACT REPORT			PAGE NO. 00001
JOBNAME	RESOURCE NAME	SCHED-ID	STEPNAME	USAGE	FREE-TYPE
PAYROLL1	PAYROLL.INCOME.DATA	001	JS30	EXC	SUCCESSFUL COMPLETION OF JOB STEP
	SYS1.PAYROLL.DATA	002		SHR	ALWAYS FREE RESOURCE
	SYSX.TEST.SAMPDATA	000	ACCNT	EXC	FREE RESOURCE MANUALLY
	SYS2.TABLE.DATA	000	JS60	SHR	ALWAYS FREE RESOURCE
	SYS3.VAULT.TAPE	000	UPDATE1	EXC	ALWAYS FREE RESOURCE
PAYROLL2	PAYROLL.FILES.PAYCHECK	200	NODATA	EXC	ABNORMAL TERMINATION
	SYS1.TEST1.DATA	000	JS10	SHR	ALWAYS FREE RESOURCE
	SYS1.TEST2.DATA	000		EXC	ALWAYS FREE RESOURCE
	SYS2.DDE.TEST.DATA	000	INCOME10	SHR	ABNORMAL TERMINATION
	SYSX.TEST.PROCS	000		SHR	ALWAYS FREE RESOURCE
PAYROLL3	SYS4.USERS.DATA	000		SHR	ALWAYS FREE RESOURCE
	CAI.TEST.DATASET	000	JS100	EXC	SUCCESSFUL COMPLETION OF JOB STEP
	SYS1.VARS.SMAP.DASD.CHECKS	000		SHR	SUCCESSFUL JOB COMPLETION
	TL6642.TEST.DATA.PAY3	238	JS20	EXC	SUCCESSFUL COMPLETION OF JOB STEP
	SYSX.PROCS.USER.DATA	000	JS30	SHR	ALWAYS FREE RESOURCE
PAYROLL4	SYS1.TEST.BENCH.DATA	000		EXC	ABNORMAL TERMINATION
	TL6642.PDS.TEST.DATA	238	JS1000	EXC	ABNORMAL TERMINATION
	SYS4.TEMP.FILEX	000		SHR	FREE RESOURCE MANUALLY
	HELP.TEMP.DATASET1	000	JS10	EXC	SUCCESSFUL COMPLETION OF JOB STEP
	HELP.TEMP.DATASET2	000	JS20	EXC	SUCCESSFUL COMPLETION OF JOB STEP
PAYROLL6	SYS1.DUMPIT.DATA.TEMP.DATA	000		SHR	ABNORMAL TERMINATION
	SYSX.TL6642.HELP	000	JS30	SHR	FREE RESOURCE MANUALLY
	SYS2.SYS2.SYS2.SYS2	000		EXC	ABNORMAL TERMINATION
	CICSREG1	000	STEP0099	ASX	FREE RESOURCE MANUALLY
	CICSREG8	000		ASX	RESOURCE MUST BE INACTIVE
PAYROLL7	DB2	000		CRQ	RESOURCE MUST BE INACTIVE
TOTAL JOB RECORDS EXTRACTED 00006					

Figure 13-4. SASSDT60-01 VRM Job Extract Report

13.14.1 VRM DBT Job Extract JCL

The first step in Virtual Resource Management Database Transportability is execution of the SASSDT60 program. See the following JCL example for this step:

```
//jobname JOB .....
//VRMDBT EXEC PGM=SASSDT60,REGION=1024K,
//      PARM='ALLJOBS'
//STEPLIB DD DISP=SHR,
//      DSN=ca7.loadlib
//CA7RSRC DD DISP=SHR,
//      DSN=ca7.vrm.database
//DT60ADDS DD DSN=ca7.vrmdbt.DT60ADDS,
//      DISP=(NEW,CATLG,DELETE),
//      UNIT=SYSDA,
//      SPACE=(CYL,(5,2),RLSE),
//      DCB=(RECFM=FB,LRECL=80,BLKSIZE=4000)
//SYSUDUMP DD SYSOUT=*
//JOBREPT DD SYSOUT=*
//SYSIN DD *
JOB=PAY*
JOB=T164000A
JOB=S10PAY99
/*
//
```

Figure 13-5. SASSDT60 Execution JCL

13.14.1.1 PARM Keywords

The SASSDT60 program accepts two PARM values. The first parameter, REPORT indicates that this is a report run only. The DT60ADDS file will not be generated; however, a report will be produced. The SASSDT60-01 VRM Job Extract Report displays the jobs and resource connections for the CA-7 Batch Terminal Interface commands that would have been generated if this had been an actual run.

The second parameter, ALLJOBS indicates that all jobs and their associated resource connections will be extracted for this run. This allows bypassing the selection of specific jobs using the SYSIN DD control cards. If the ALLJOBS PARM is coded, any control cards found in the SYSIN DD are ignored.

```
//jobname job .....
// EXEC PGM=SASSDT60,PARM='REPORT'

//jobname job .....
// EXEC PGM=SASSDT60,PARM='REPORT,ALLJOBS'
```

Figure 13-6. PARM Examples

13.14.1.2 SYSIN Control Cards

The SYSIN DD * file in the JCL allows selection of a specific job from the VRM database component. The keyword JOB= is used to specify a fully qualified or generic job name. A generic job name can be specified by coding an asterisk (*) on the trailing end of the qualifying job name character(s). At least one character is required. Comments may be included in the SYSIN DD file by coding an asterisk in the first column (col 1). There is no limit to the number of control cards that can be specified.

```
//SYSIN  DD *
*
* Comment card (asterisk column 1)
*
* The following control card selects any job with the
* first three characters of PAY.
JOB=PAY*
*
* The following control card selects job TESTJOB1 only.
JOB=TESTJOB1
*
* The following control cards selects any job which begins
*with the letter Z.
JOB=Z*
```

Figure 13-7. SYSIN DD Control Card Examples

13.14.1.3 Data Sets Used/Created

CA7RSRC This is the CA-7 VRM database component.

DT60ADDs This is the generated CA-7 Batch Terminal Interface commands file. This file is used as input into a CA-7 BTI job to add the job to resource connections on a different VRM database.

SYSIN This file contains the control cards used to specify which jobs will be extracted for this run.

JOBREPT This is the SASSDT60-01 VRM Job Extract Report file.

Note: The second step of the VRM Database Transportability process is to run a CA-7 Batch Terminal Interface job using the generated DT60ADDs file as SYSIN input to add the job to resource connections on the target VRM database.

The SASSDT60 program does not generate commands to define the number of available resource count resources at the receiving site. This must be done manually using the RM.7 screen.

13.15 Automated Recovery Facility Database Extracts

The Automated Recovery Facility (ARF) Database Transportability facility is used to extract ARFSET information from the ARF database and generates CA-7 Batch Terminal Interface commands, in a fixed file format, for transport to a different database.

The SASSDT70 program is used to read the ARF database and extract ARFSET information based on user-specified criteria. The program creates a DT70ADDS data set that contains the CA-7 Batch Terminal interface commands required to add the ARFSET definitions to a different ARF database component. Besides the DT70ADDS files, an ARFSET selection report SASSDT72-01 is generated.

The first step in the ARF database transportability process is to extract ARFSETs from the database. See the following JCL example for this step:

```
//JOBNAME JOB ACCOUNTING INFO,PROGRAMMER,CLASS=A,MSGCLASS=A
//*
//JS10 EXEC PGM=SASSDT70,PARM='REPORT,ALLSETS'
//STEPLIB DD DISP=SHR,DSN=CAI.CA7.LOADLIB
//*
//CA7ARFDB DD DISP=SHR,DSN=CAI.CA7.ARF
//ARFREPT DD SYSOUT=*
//DT70ADDS DD DSN=USER.DT70ADD,
// DISP=(NEW,CATLG,DELETE),
// UNIT=SYSDA,
// SPACE=(TRK,(1,1),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=4000)
//SYSIN DD DUMMY
```


13.15.1 PARM Keywords

The SASSDT70 program accepts two parameter values. The first parameter, REPORT, indicates that this is a report run only. The DT70ADDS file will not be generated; however, a report will be produced. The SASSDT72-01 ARFSET Extract Report displays the ARFSET definitions for the CA-7 Batch Terminal Interface commands that would have been generated if this had been an actual run.

The second parameter, ALLSETS indicates that all ARFSETs will be extracted for this run. This allows bypassing the selection of specific ARFSETs using the SYSIN DD control cards. If the ALLSETS PARM is coded, any control cards found in the SYSIN DD are ignored.

```
//jobname job .....
// EXEC PGM=SASSDT70,PARM='REPORT'

//jobname job .....
// EXEC PGM=SASSDT70,PARM='REPORT,ALLSETS'
```

The SYSIN DD * file in the JCL allows selection of a specific ARFSET from the ARF database component. The keyword ARFSET= is used to specify a fully qualified or generic ARFSET name. A generic ARFSET name can be specified by coding an asterisk (*) on the trailing end of the qualifying set name character(s). At least one character is required. Comments may be included in the SYSIN DD file by coding an asterisk in the first column (col. 1). The number of control cards that can be specified is limited only by the amount of memory available to the program.

```
//SYSIN  DD *
*
* Comment card (asterisk column 1)
* The following control card selects any ARFSET with the
* first three characters of PRD.
ARFSET=PRD*
*
* The following control card selects ARFSET TEST only.
ARFSET=TEST
*
* The following control cards selects any ARFSET which begins
*with the letter Z.
ARFSET=Z*
```

13.15.2 Data Sets Used/Created

CA7ARFDB

This is the CA-7 ARF database component.

DT70ADDS

This is the generated CA-7 Batch Terminal Interface commands file. This file is used as input into a CA-7 BTI job to add the ARFSET definitions to a different ARF database.

SYSIN

This file contains the control cards used to specify which ARFSETs will be extracted for this run.

ARFREPT

This is the SASSDT72-01 ARFSET Extract Report file.

Note: The second step of the ARF Database Transportability process is to run a CA-7 Batch Terminal Interface job using the generated DT70ADDS file as SYSIN input to add the ARFSET definitions on the target ARF database.

Index

Special Characters

statements

See ?

A

Abends 3-7

Access restrictions 1-10

Accessing files 10-5

Active area

 appending JCL card-image data 10-3

 defined 11-1, 11-3

 saving documentation 7-2

 saving schedule IDs 3-15

Active job resources display screen 5-15

ADD function, alias 1-20

Adding

 application data sets at new site 13-22

 base calendars (DB.2.8 screen) 3-55

 CA-Librarian JCL at new site 13-26

 CA-Panvalet JCL at new site 13-29

 cataloged procedures at new site 13-32

 CPU job at new site 13-13

 data set

 level documentation member to database 7-16

 requirements 4-8

 to database 9-2, 9-6

 DD level documentation member to database 7-20

 documentation at new site 13-34

 input network

 schedule to database 3-25

 SCHID to work area 3-28

 job

 level documentation member to database 7-5

 schedule member to database 3-12

 SCHID to work area 3-15

 to database 2-2, 2-4

 network

 at new site 13-19

 level documentation member to database 7-8

 to database 8-2, 8-4

 output network

 schedule to database 3-37

 SCHID to work area 3-40

 PDS

 JCL members at new site 13-24

 member containing JCL 10-4

Adding (*continued*)

 schedules at new site 13-36

 system level documentation member to database 7-24

 user level documentation member to database 7-12

Address space resources 5-2

Adjusting schedule days 3-16, 3-29

Aliases 1-20, 1-23

 for formatted screen functions 1-20, 1-23

Altering resolved schedule information for job or input

 network 3-52

Alternate JCL libraries 10-31

Analyze

 application 13-6

 commands 12-6

APA reporting 1-13

APPEND function, alias 1-20

Appending

 data set level documentation data 7-16

 DD level documentation data 7-20

 documentation data 7-24

 JCL card-image data in Active Area 10-3

 job level documentation data 7-5

 network level documentation data 7-8

 user level documentation data 7-12

Application data sets 13-22

Application System Documentation screen 7-23

AR.3 screen 6-10

ARF (Automated Recovery Facility)

 AR.3 screen 6-10

 ARF Condition Definition Edit screen 6-12

 ARF statement 10-14

 overview 1-9, 6-1

ARF statement 10-13

ARFSETs

 defining 6-2

 overriding 10-14

Automated Performance Analysis (APA) reporting 1-13

Automated Production Control system 1-6

B

Base calendars

 and database transportability 13-4

 defined 3-2, 3-3

Batch Card Load Program (BCLP) 1-11

Batch input 1-22

Batch Terminal Interface (BTI)

See BTI

BCLP

- overview 1-11

BTI

- add data set to database 9-2
- add job to database 2-2
- add network to database 8-2
- Analyze commands 12-6
- change network characteristics 8-6
- changing job's characteristics 2-16
- creating documentation commands 7-2

Bypassing

- menus/screens 1-18, 1-22
- normal scheduling activities 3-8
- primary menus 1-14

C

CA-11

- considerations 1-10
- Step insertion 13-4

CA-Dispatch and demand networks 3-38

CA-Driver 1-11

CA-Earl reporting 1-13

CA-Easytrieve Plus reporting 1-13

CA-JCLCheck 10-27

CA-Librarian JCL

- database transportability 13-26
- maintaining 10-2
- Member Adds Report (SASSDT30-06) 13-27
- Member Deletes Report (SASSDT30-07) 13-28
- reviewing 10-2

CA-OPS/MVS II 5-10

CA-Panvalet JCL

- database transportability 13-29
- maintaining 10-2
- Member
 - Adds Report (SASSDT30-08) 13-30
 - Deletes Report (SASSDT30-09) 13-31

CA7RSRC 13-55

Calendar macro 3-3

Calendar schedules 13-4

Calendar-based schedule 8-2

CALMOD

- See* DB.2.8 screen

Cataloged procedures 13-5, 13-32

Changing

- alias 1-20
- data set
 - characteristics 9-10
 - information in database 9-6
 - name in database 9-6, 9-7
- DCB attribute 9-10

Changing (*continued*)

- JCL 2-16

job

- characteristics 2-16
- information in database 2-5
- network characteristics 8-6
- processing days 3-52
- requirements 4-3
- schedule 3-5
- screens 1-20

CLEAR function, alias 1-20

Clearing screen

- of all input data 2-5
- of input data and reset ID-COUNT 7-12
- of input data and reset SCHID-COUNT 3-12

Command data sets 13-47

Commands

DEMAND

- running jobs by request 3-7

DEMANDH 3-7

DMDNW

- network definition requirements 8-2
- overview 3-8

DSN 9-6

JCLxx 10-12

JOB 2-4

JOBCONN 4-6

JOBCONN,DSN 4-9

JOBCONN,JOB 4-14

JOBCONN,NWK 4-20

JOBCONN,RPT 4-30

JOBCONN,USR 4-26

LIST 7-36

LJCK 10-12

LOAD 10-28

LOADH 10-28

LPROS

- defining documentation 7-1
- LPROS loop 7-25
- with RESTART 7-36

NETWORK 8-4

PRINT 3-3

PROS,DD 7-20

PROS,DSN 7-16

PROS,JOB 7-5

PROS,NWK 7-8

PROS,USER 7-12

PROSE 7-3

PROSE,SYS 7-24

RESOLV 3-4

RUN 3-8

- bypassing normal scheduling 3-8

Commands (*continued*)

- RUNH 3-8
- RUNNW 3-8
- SCHD 3-9
- SCHD,DTRG 3-46
- SCHD,INWK 3-25
- SCHD,JOB 3-12
- SCHD,JTRG 3-44
- SCHD,NTRG 3-45
- SCHD,ONWK 3-37
- SCHDMOD 3-52

Commas in data 13-6

Common formatted screen fields 1-15

Condition code tests 10-18

Conditions for adding job 2-2

Control statements

- "#ARF" 10-14
- "#JEND" 10-10
- "#JI" 10-7
- "#JO" 10-7
- "#MSG" 10-16
- "#RES" 10-17
- "#SCC" 10-18
- "#XEND" 10-10
- "#XI" 10-10
- "#XO" 10-10

Control statements, reserved 10-6

Corequisite resources 5-2

Corequisite resources list screen 5-23

CPM 5-10

CPU

- Job Definition screen 2-3
- job dependency relationships 2-16
- Job Documentation screen 7-4
- Job Predecessors screen 4-13
- job schedules 13-36, 13-40
- Job Scheduling Parameter Edit commands 3-15
- Job Scheduling Parameter Edit screen 3-14
- Job Scheduling screen 3-11

Creating

- command files 13-41
- processing schedules for
 - jobs scheduled on date/time basis 3-4
 - workstation networks scheduled on date/time basis 3-4
- temporary JCL 10-29, 10-31

Criteria for scheduling 3-14

Critical Path Monitoring 5-10

D

Data Set

- Definition screen 9-5
- Documentation screen 7-15
- Predecessors screen 4-8
- Triggering screen 3-46

Data sets

- adding 9-1
- allocation 1-10
- connections 4-12
- defined as preexecution requirement 9-4
- dynamically allocated 9-4
- level documentation 7-15
- maintaining 9-1
- triggers 3-6, 13-37

Database

- Adds Part 1 report (SASSDT30-01) 13-14
- Adds Part 2 report (SASSDT30-02) 13-16
- Deletes report (SASSDT30-03) 13-17
- Disables report (SASSDT30-04) 13-18
- extraction 13-8
- Maintenance Menu 1-17
- resynchronizing with new JCL 2-16
- transportability
 - overview 13-1
 - special considerations 13-50
- verification 12-1

Date and time stamp 1-15

Date/time-driven scheduling 3-2, 3-3

DB

- command 1-17
- commands 1-14
- function, alias 1-20
- menu screens 1-14, 1-17

DB.1 screen

- adding job to database 2-2
- changing job's characteristics 2-16
- entering CPU job data 2-3
- JCL change made outside CA-7 2-16
- job marked MAINT=Y 4-12

DB.2 screen 3-9

DB.2.1 screen

- defining
 - options for CPU jobs 3-11

DB.2.2 screen

- defining calendar-based schedule 8-2
- scheduling options for input networks 3-24

DB.2.3 screen

- defining options for output network schedules 3-36
- entering schedule parameters 8-2

- DB.2.4 screen
 - defining job triggers 3-6, 3-44
 - reviewing job triggers 3-44
- DB.2.5 screen, defining network triggers 3-6
- DB.2.6 screen, defining data set triggers 3-6
- DB.2.7 screen
 - altering resolved schedule information 3-52
 - changing existing schedule 3-5
 - modifying resolved dates 3-52
- DB.2.8 screen 3-55
- DB.3 Menu screen 4-6
- DB.3.1 screen
 - adding data set requirements 4-8
 - defining data set as preexecution requirement 9-4
 - disconnecting old data set name 9-10
 - maintaining data set/job connections 9-9
 - modifying data set requirements 4-8
- DB.3.2 screen, establishing
 - dependency relationships for CPU job 2-16
 - predecessor job requirements 4-13
- DB.3.3 screen
 - adding network to database 8-2
 - defining input network connection 8-2
- DB.3.4 screen
 - connecting network to CPU job 3-38
 - defining
 - job requirements for networks 4-19
 - output network connection 8-2
 - redefining job connection 8-6
- DB.3.6 screen 4-25
- DB.3.7 screen 4-29
- DB.4 Menu screen 7-3
- DB.4.1 screen 7-4
- DB.4.2 screen 7-7
- DB.4.3 screen 7-11
- DB.4.4 screen 7-15
- DB.4.5 screen 7-19
- DB.4.6 screen 7-23
- DB.5 screen
 - adding network to database 8-2
 - changing network characteristics 8-6
 - defining networks 8-3
- DB.6 screen
 - adding data set to database 9-2
 - data set TYPE=PERM 4-12
 - defining
 - data set triggers 3-6
 - dynamically allocated data set 9-4
 - external data set 9-4
 - permanent data set 9-4
 - user data set 9-5
 - marking data set as permanent for all jobs 4-11

- DB.6 screen (*continued*)
 - updating
 - data set characteristics 9-10
 - data set information in database 9-6
- DB.7 screen 2-16, 4-3, 10-2
- DBMADDS1 report 13-34
- DD
 - function 2-4
 - level documentation 7-19
 - Statement Documentation screen 7-19
- Deadline prompting 2-13
- DEFAULTS job record 2-2, 2-15
- Defining
 - alternate library 10-31
 - annual schedule 3-20, 3-33
 - calendar-based schedule 8-2
 - connection 4-6
 - daily schedule 3-17, 3-30
 - data set
 - level documentation 7-15
 - triggers 3-6
 - data sets
 - which trigger job scheduling 3-46
 - with DB.6 screen 9-5
 - database 1-17
 - DD level documentation 7-19
 - documentation 7-1
 - dynamically allocated data set 9-4
 - individual workstation schedules in output network 3-39
 - input network connection 8-2
 - input networks which trigger job scheduling 3-45
 - job
 - level documentation 7-4
 - requirements for networks 4-19
 - triggers 3-6
 - jobs 2-2
 - jobs which trigger scheduling of other jobs 3-44
 - monthly schedule 3-18, 3-31
 - network
 - level documentation 7-7
 - triggers 3-6
 - options taken for
 - CPU jobs with date/time schedules 3-11
 - output network schedules 3-36
 - output network connection 8-2
 - predecessor requirement 4-6
 - requirements 4-2
 - schedules for jobs run on date and time basis 3-14
 - scheduling
 - criteria 3-14
 - options for input networks 3-24
 - parameters for input networks scheduled on date/time basis 3-27

Defining (*continued*)

- special override library 10-29
- step level condition code tests 10-18
- symmetric schedule 3-21, 3-34
- system level documentation 7-23
- systems 2-3
- triggers 3-44
- user level documentation 7-11
- weekly schedule 3-17, 3-30

DELETE function, alias 1-20

Deleting

- application data sets at original site 13-23
 - base calendars with DB.2.8 3-55
 - CA-Librarian JCL at original site 13-28
 - cataloged procedures at original site 13-33
 - data set
 - and its elements from database 9-6
 - level documentation member from database 7-16
 - DD level documentation member from database 7-20
 - documentation
 - at original site 13-35
 - member from database 7-24
 - input network
 - schedule member from database 3-25
 - SCHID from work area 3-28
 - job
 - and its data from database 2-4
 - and its data sets 2-4
 - at original site 13-17
 - level documentation member from database 7-5
 - schedule member from database 3-12
 - SCHID from work area 3-15
 - member of PDS containing JCL 10-3
 - network
 - and its documentation/schedules from database 8-4
 - at original site 13-20
 - level documentation member from database 7-8
 - output network
 - schedule member from database 3-37
 - SCHID from work area 3-40
 - PDS JCL members at original site 13-25
 - schedules at original site 13-40
 - user level documentation member from database 7-12
- ## DELPRRN function, deleting JCL in trailer queue 2-5
- ## DEMAND command
- adding jobs 2-2
 - and TYPE=RES 4-12, 4-24
 - running jobs by request 3-7
- ## DEMANDH command
- running jobs by request 3-7

Dependence definition 4-2

Dependency relationships for CPU job 2-16

Device control, VRM 5-28

Diagnostic messages 1-15

Directly connected 11-30

Disabling

- CA-Panvalet JCL at original site 13-31
- job at original site 13-18
- network at original site 13-21

Disconnecting old data set name 9-10

Discontinuing job 2-11

DL parameter value 10-9

DMDNW command

- network definition requirements 8-2
- overview 3-8

Documentation

- adding to database 13-34
 - categories 7-2, 7-26
 - defining
 - segment names 7-27
 - subsegment names 7-29
 - displaying
 - segments 7-28
 - subsegments 7-31
 - format of
 - segments 7-27
 - special purpose 7-33
 - subsegments 7-30
 - LINKed documentation 13-7
 - naming conventions 7-32
 - overview 1-12
 - screens 7-1
 - segment names
 - definition 7-27
 - reserved 7-26
 - segment usage
 - END 7-37
 - QDSEC 7-34
 - RESTART 7-36
 - station-name 7-35
 - segmentation guidelines 7-26
 - special
 - features 7-26
 - purpose 7-33
 - subsegments
 - displaying 7-29
 - guidelines 7-29
 - user-defined segments 7-26
 - workload 7-1
- ## DSN
- command 9-5, 9-6
 - connections 13-16

- DSNBR (CA-7 assigned) 9-2
- DT60ADDS file 13-53, 13-55
- Due-out
 - day for network workstation 3-30
 - time of day
 - network workstation 3-29, 3-41
 - schedule ID 3-16
 - TRGD-JOB 3-48
- Dynamically allocated data set 9-4

E

- Edit facility 11-1
- EDIT function
 - alias 1-20
 - DB.2.1 screen 3-12
 - DB.2.2 screen 3-25
 - DB.2.3 screen 3-37
 - default requests 11-32
- Edit screen
 - CPU Job Scheduling Parameter Edit 3-14
 - Input Network Scheduling Parameter Edit 3-27
 - Output Network Scheduling Parameter Edit 3-39
- Edit Work File (EWF)
 - defined 11-1
 - validating 10-27
- Editing
 - data set level documentation 7-16
 - DD level documentation 7-20
 - JCL in PDS member 10-3
 - job level documentation 7-5
 - limitations 11-34
 - network level documentation 7-8
 - system level documentation 7-24
 - user level documentation 7-12
- Editor 11-6
- Elapsed
 - lead time for TRGD-JOB 3-49
 - queue time of TRGD-JOB 3-49
- Enhancements 1-1
- Event scheduling 3-6
- Event-driven scheduling 3-2
- Exception conditions and ARF 6-2
- Exclusive resource 5-2
- Execution
 - JCL libraries 13-5
 - requirements 4-2
- EXIT function, returning to
 - DB.2.1 screen and restore data 3-15
 - DB.2.2 and restore data 3-28
 - DB.2.3 and restore data 3-40

- Exiting
 - database maintenance mode 1-14
 - from DB using batch input 1-23
- Expansion of Requested Jobs Report (SASSDT20-01) 13-9
- Express priority 2-14
- External Communications Facilities 1-11
- External data set 9-4

F

- FE function
 - alias 1-20
 - DB.2.1 screen 3-12
- FETCH and EDIT
 - data set level documentation 7-16
 - DD level documentation 7-20
 - input network schedule data 3-25
 - JCL in PDS member or sequential data set 10-3
 - job level documentation 7-5
 - job schedule 3-12
 - network level documentation 7-8
 - output network schedule 3-37
 - system level documentation 7-24
 - user level documentation 7-12
- FEM (Full Edit Mode)
 - command descriptions 11-11
 - commands 11-11
 - sample screens 11-19
 - screen areas 11-9
 - subcommands 11-10
- FETCH function
 - alias 1-20
 - DB.2.1 screen 3-12
 - retrieving
 - data set level documentation 7-16
 - DD level documentation 7-20
 - input network schedule data from database 3-25
 - JCL data 10-3
 - job level documentation 7-5
 - job schedule data from database 3-12
 - network level documentation 7-8
 - output network data from database 3-37
 - system level documentation 7-24
 - user level documentation 7-12
- FETCHP function, alias 1-20
- Fields common to all formatted screens 1-15
- Flows 5-10
- FORMAT function
 - alias 1-20
 - DB.1 screen 2-5
 - DB.2.7 screen 3-53

- Formatted screens
 - common fields 1-15
 - functions/alias names 1-20
 - using for online input 1-14
- Full Edit Mode
 - See* FEM (Full Edit Mode)
- Function
 - ADD 2-4
 - alias names 1-20
 - APPEND 7-5
 - DD 2-4
 - DELETE 2-4
 - DELPRRN 2-4
 - EDIT 3-12
 - EXIT 3-12
 - FE 3-12
 - FETCH 3-12
 - field 1-14
 - FORMAT 2-4
 - LIST 2-4
 - Menu screen 1-16
 - PURGE 2-4
 - RENAME 9-6
 - REPL 3-12
 - RESOLV 3-12
 - RUN 10-3
 - RUNH 10-3
 - SAVE 3-12
 - shortcuts 1-19
 - SR 3-12
 - SS 3-12
 - transfer 1-20, 1-23
 - UPD 2-4

G

- GDG 9-7
- Generation data group
 - See* GDG

H

- HELP facility 1-7
- HELP library 2-6
- History reporting 1-13
- HLD statement 10-13
- HOLD field 2-7

I

- IEHPRGM-type functions 1-10

- In-stream JCL procedures 13-6
- Including
 - a job 10-7
- Initialization file
 - and JCL library 10-4
 - DAIO statement 11-3
 - DEFAULTJOB parameter 2-15
 - parameters for
 - date and time scheduling 3-3
 - schedule scan 3-6
 - STATIONS statement 7-35
- Input
 - screens 1-14
 - workstation networks 13-21
- Input Network
 - database transportability 13-38, 13-40
 - defining 8-2, 8-3, 8-4
 - Scheduling Parameter Edit commands 3-28
 - Scheduling Parameter Edit Screen 3-27
 - Scheduling screen 3-24
 - Triggering screen 3-45
- Input/Output Network
 - Definition screen 8-3
 - Documentation screen 7-7
 - Tasks screen 4-19
- Interfaces
 - to external security packages 1-10
 - TSO/ISPF 11-30
- ISPF
 - connected 11-30
 - Editor 11-30
 - interface 11-30
 - SUBMIT 11-34

J

- J statements 10-6
- JCL
 - changes 2-16, 4-3
 - command 10-2, 10-3
 - libraries
 - alternate 10-31
 - identification 10-4
 - member 10-24
 - scanning with sample JCL 10-26
 - Library Maintenance screen 10-2
 - management 10-1
 - overrides 1-11, 10-6
 - screens 10-2
 - special override library 10-29
 - statement 10-13
 - statements (using DB.7 screen) 10-2

- JCL (*continued*)
 - syntax checking 10-27
 - temporary 10-29, 10-31
 - utility 10-24
 - validation 10-27
- JCL-OVRD field 2-7
- JCLCHECK 10-27
- JCLID field 2-6
- JCLOVRD command 2-7
- JCLxx command 10-12
- JEND statement 10-10
- JI statement 10-7
- JO statement 10-7
- Job
 - class 2-13
 - command 2-3, 2-4
 - connections 13-15
 - defining CPU jobs 2-3
 - Definition screen 2-3
 - dependency, predecessor job requirements 4-13
 - discontinuation of 2-11
 - field 2-5
 - flow control
 - See* Work flow control
 - include 10-7
 - level documentation 7-4
 - message, routing to logical terminal 2-12
 - name UCC7Rxxx, for workload balancing 2-5
 - omit 10-7
 - predecessor/dependency relationship 8-2
 - Predecessor/Successor Menu 4-6
 - priority 2-14
 - requirements 4-13
 - resource management screen 5-6
 - restart 1-10
 - schedule, FETCH and EDIT 3-12
 - Scheduling Parameter Edit commands 3-15
 - Scheduling Parameter Edit screen 3-14
 - Scheduling screen 3-11
 - screen 2-3
 - timing, start of 2-11
 - Triggering screen 3-44
 - triggers 3-6, 13-37
- Job/network connections 8-6
- Job/resource cross reference list screen 5-12
- JOBCONN command 4-6
- JOBCONN,DSN command 4-9
- JOBCONN,JOB command 4-14
- JOBCONN,NWK command 4-20
- JOBCONN,RPT command 4-30

- JOBCONN,USR command 4-26
- JOBREPT file 13-55
- Jobs
 - defining 2-2
 - dependency 4-2
 - documentation 7-1
 - predecessor 4-2
 - Requested To Be Moved Report (SASSDT10-01) 13-8
 - scheduling 3-2
 - screen 2-3
 - separating 4-13
 - successor 4-2
 - waiting on resources screen 5-20

L

- Lead time
 - for schedule ID 3-17, 3-42
 - for TRGD-JOB 3-49
 - for workstation 3-29
- LINKed documentation 13-7
- List
 - function alias 1-20
 - function shortcuts 1-19
- LIST command 7-36
- Listing
 - base calendars 3-3, 3-55
 - data set level documentation 7-16
 - DD level documentation 7-20
 - job names and schedule IDs for triggering element 3-47
 - schedule information from database 3-53
 - SCHIDs for
 - input network 3-28
 - job 3-15
 - output network 3-40
 - system level documentation 7-24
- LJCK command 10-12
- LNTWK command
 - and database transportability 13-19
 - using when changing a network 8-6
- LOAD command
 - adding jobs 2-2
 - changing jobs 2-16
 - creating job profile data 10-28
- Load process
 - adding data sets 9-2
 - and changing JCL 2-16
 - and dynamically allocated data sets 9-4
 - and execution requirements 4-2
- LOAD/RELOAD status 13-4
- LOADH command 2-13, 10-28

LPROS command

loop

DB.4.3 screen 7-13

DB.4.4 screen 7-17

DB.4.5 screen 7-21

DB.4.6 screen 7-25

with RESTART 7-36

M

Maintaining data set/job connections 9-9

Management level reporting 1-13

Master station 10-16

Menu

bypassing secondary 1-18

DB (database maintenance) 1-17

documentation 7-3

function 1-16

Job Predecessor/Successor 4-6

RM (resource management) 5-5

Scheduling 3-9

VRM 5-5

MESSAGE

field 1-15

routing to logical terminal 2-12

MIXED FEM subcommand 11-16

MNT statement 10-13

Modification to Resolved Schedule Dates screen 3-52

Modifying

alias 1-20

data set requirements 4-8

processing schedules for

jobs scheduled on date/time basis 3-4

workstation networks scheduled on date/time

basis 3-4

resolved schedule dates for job or input network 3-52,
3-55

schedule 3-5

Month-by-month listing of base calendar(s) 3-3

MSG statement 10-13, 10-16

MSG-INDX field 1-15

MSGs field 1-15

Mutual exclusion 4-16

N

Naming conventions for documentation 7-32

Network

and job connections 8-6

command 8-3, 8-4

connections 8-2

defining 8-1

Network (*continued*)

deleting 8-3

demanding 3-7

documentation 7-3

level documentation 7-7

overview 8-1

scheduling 3-9

screen 8-2

triggers 3-6, 13-37

NORM-type data set 9-7

NOX statement 10-13

NTR statement 10-13

NWK connections 13-15

NXTCYC settings 13-5

O

Omit job 10-7

On-request scheduling 2-2, 3-7

Online

assistance 1-7

documentation 7-1

input 1-14

utilities 1-10

Output Network

commands when moving 13-39

defining 8-4

schedule 3-37, 13-40

Scheduling Parameter Edit commands 3-40

Scheduling Parameter Edit screen 3-39

Scheduling screen 3-36

Output workstation network 13-21

Overriding

ARFSETs 10-14

JCL 10-6

P

PA key

assignment 1-7

using with ISPF editor 11-34

using with text editor 11-29

Panel ID field 1-15

PDS JCL members 13-24

Pending resources job display screen 5-18

PERM data set 9-3, 9-7

Permanent data sets 9-3

PF key

assignment 1-7

using with ISPF editor 11-34

using with text editor 11-29

- Pointer verification 12-2
- POST AT CLOSE TIME field 3-7, 9-8
- POST command 4-5, 4-25
- Predecessor requirements
 - defining 4-12
 - dependency 4-2, 4-13
 - satisfying 4-5
 - temporary 4-4
- Predecessor/successor relationships 13-15
- Preexecution requirement 2-12, 9-4
- PRINT command 3-3
- Printing contents of JCL members in PDS 10-24
- PROCLIB Member Deletes Report (SASSDT30-05) 13-33
- Producing month-by-month listing of base calendar(s) 3-3
- Production control 1-6
- Productivity 1-13
- PROGRAM field 1-15
- Prompting messages, issuing if job late 2-12
- PROMPTS field 2-12
- PROS,DD command 7-20
- PROS,DSN command 7-16
- PROS,JOB command 7-5
- PROS,NWK command 7-8
- PROS,USER command 7-12
- PROSE command 7-3
- PROSE,SYS command 7-24

Q

- QM.1 screen
 - and posting requirements 4-5
 - indicating task completion 4-25
- QM.2 screen
 - indicating task completion 4-25
 - viewing requirements of a job 4-5
- QUIT function, alias 1-20

R

- RDAY field 3-14, 3-27
- Redefining job connection 8-6
- Relative days 3-19, 3-32
- Release enhancements summary 1-1
- Removing
 - expired CA-7 JCL Scheduled Override control statements 10-24
 - network station 8-5
 - scheduled JCL 10-24
- RENAME
 - function 9-6
 - function, alias 1-20

- Renaming
 - data set 9-6, 9-7, 9-10
 - PDS member containing JCL 10-3
- REPL function, alias 1-20
- Replacing
 - data set level documentation member 7-16
 - DD level documentation member 7-20
 - input network
 - schedule in database 3-25
 - SCHID in work area 3-28
 - job
 - level documentation member 7-5
 - schedule member in database 3-12
 - network level documentation member 7-8
 - output network SCHID in work area 3-40
 - PDS member or sequential data set containing JCL 10-3
 - schedule member in database 3-37
 - SCHID in work area 3-15
 - system level documentation member 7-24
 - user level documentation member 7-12
- Report IDs Created screen 4-29
- Reporting
 - Automated Performance Analysis 1-13
 - CA-Earl reporting 1-13
 - CA-Easytrieve Plus reporting 1-13
 - CA-Panvalet JCL Member
 - Adds Report (SASSDT30-08) 13-30
 - Deletes Report (SASSDT30-09) 13-31
 - Database Adds
 - Part 1 Report (SASSDT30-01) 13-14
 - Part 2 Report (SASSDT30-02) 13-16
 - Database Deletes Report (SASSDT30-03) 13-17
 - Database Disables Report (SASSDT30-04) 13-18
 - Expansion of Requested Jobs Report (SASSDT20-01) 13-9
 - history 1-13
 - Jobs Requested To Be Moved Report (SASSDT10-01) 13-8
 - LIBRARIAN JCL Member
 - Adds Report (SASSDT30-06) 13-27
 - Deletes Report (SASSDT30-07) 13-28
 - PROCLIB Member Deletes Report (SASSDT30-05) 13-33
 - productivity 1-13
- Requirements
 - automatically tracked and satisfied by system 4-5
 - definitions 4-2
 - needing manual intervention 4-5
 - network connections 13-15
 - posting 9-4
 - satisfied 4-25

- RES statement 10-13, 10-17
- RESANL command 2-14, 2-15, 12-6
- Reserved DDname Table 9-3
- RESOLV
 - command 3-4
 - function
 - overview 3-4
 - using with DB.2.1 screen 3-12
 - using with DB.2.2 screen 3-25
- RESOLV function alias 1-20
- Resource count resource management screen 5-25
- Resource count resources 5-3
- Resources
 - address space 5-2
 - corequisite 5-2
 - exclusive 5-2
 - handling 5-4
 - resource count 5-3
 - shared 5-2
- Restricting access 1-10
- Resynchronizing database with new JCL 2-16
- Retrieving
 - data set level documentation 7-16
 - DD level documentation 7-20
 - input network schedule data from database 3-25
 - JCL data 10-3
 - job
 - level documentation 7-5
 - schedule data from database 3-12
 - network level documentation 7-8
 - output network data from database 3-37
 - system level documentation 7-24
 - user level documentation 7-12
- Returning
 - to DB.2.1 screen and restoring data 3-15
 - to DB.2.2 screen and restoring data 3-28
 - to DB.2.3 screen and restoring data 3-40
- Reviewing
 - CA-Librarian JCL 10-2
 - CA-Panvalet JCL 10-2
 - CPU job data 2-3
 - data sets which trigger job scheduling 3-46
 - individual workstation schedules in output network 3-39
 - input networks which trigger job scheduling 3-45
 - JCL statements
 - in PDS 10-2
 - in sequential JCL library 10-2
 - jobs which trigger scheduling of other jobs 3-44
 - options taken for
 - CPU jobs with date/time schedules 3-11
 - output network schedules 3-36
 - schedules for jobs run on date and time basis 3-14

- Reviewing (*continued*)
 - scheduling
 - options for input networks 3-24
 - parameters for input networks scheduled on date and time basis 3-27
 - RM Menu screen 5-5
 - RM.1 screen 5-6
 - RM.2 screen 5-12
 - RM.3 screen 5-15
 - RM.4 screen 5-18
 - RM.5 screen 5-20
 - RM.6 screen 5-23
 - RM.7 screen 5-25
 - ROLL field
 - defining/reviewing schedules 3-14
 - defining/reviewing schedules for input networks 3-27
 - Rolling schedule
 - back 3-16, 3-29
 - forward 3-16, 3-29
 - Routing message about job to logical terminal 2-12
 - RPT (Reports Processing and Tracking)
 - and demand networks 3-38
 - RQMT command 12-6
 - RQVER command 12-6
 - RSRC keyword 5-10, 5-27
 - RUN command
 - and true predecessor requirements 4-12, 4-24
 - bypassing normal scheduling 3-8
 - RUN function 10-3
 - RUNH command
 - assigned to class 9 2-13
 - bypassing normal scheduling 3-8
 - RUNH function on DB.7 screen 10-3
 - RUNNW command 3-8, 8-2

S

- SASSBSTR program 13-44
 - producing listings 13-44
- SASSDT10
 - execution JCL 13-43
 - module 13-44
- SASSDT10-01 report 13-8, 13-43
- SASSDT20
 - execution JCL 13-45
 - module
 - CA-Librarian JCL 13-26
 - CA-Panvalet JCL 13-29
 - cataloged procedures 13-32
 - expanding requested job 13-9
 - PDS JCL members 13-24

- SASSDT20-01 report 13-9, 13-45
- SASSDT30
 - execution JCL 13-49
 - module 13-10, 13-46, 13-47
- SASSDT30-01 report
 - adding at new site
 - application data sets 13-22
 - documentation 13-35
 - PDS JCL members 13-24
 - workstation networks 13-19
 - sample report 13-14
- SASSDT30-02 report
 - adding schedules at new site 13-36, 13-38, 13-39
 - sample report 13-16
- SASSDT30-03 report 13-17, 13-35
- SASSDT30-04 report 13-18
- SASSDT30-05 report 13-33
- SASSDT30-06 report 13-27
- SASSDT30-07 report 13-28
- SASSDT30-08 report 13-30
- SASSDT30-09 report 13-31
- SASSDT60
 - execution JCL 13-54
 - module 13-53
- SASSDT60-01 report 13-53
- SASSDTAB table 5-28
- SASSJCLU batch utility 10-24
- SASSMSG module 7-34, 7-35
- SASSPMDD table 9-3
- SASSUTBL module 2-14
- Satisfying requirements 4-5
- SAVE function
 - adding
 - data set level documentation member to database 7-16
 - DD level documentation member to database 7-20
 - input network schedule to database 3-25
 - job level documentation member to database 7-5
 - job schedule member to database 3-12
 - job SCHID-COUNT and SCHID in work area 3-15
 - network level documentation member to database 7-8
 - output network schedule to database 3-37
 - PDS member containing JCL 10-4
 - system level documentation member to database 7-24
 - user level documentation member to database 7-12
 - alias 1-20
 - replacing sequential data set containing JCL 10-4
 - updating
 - input network SCHID-COUNT and SCHID in work area 3-28
 - output network SCHID-COUNT and SCHID in work area 3-40
- SCAL field 3-13
- SCC statement 10-13, 10-18
- SCHD command 3-9
- SCHD,DTRG command 3-46
- SCHD,INWK command 3-25
- SCHD,JOB command 3-12
- SCHD,JTRG command 3-44
- SCHD,NTRG command 3-45
- SCHD,ONWK command 3-37
- SCHDMOD command 3-52
- Schedule
 - adding at new site 13-36
 - defining 3-4
 - deleting at original site 13-40
 - ID, importance of 3-2
 - modifications 3-5, 13-4
 - resolving 3-4
 - triggers
 - data set 3-6
 - job 3-6
 - network 3-6
- Schedule scan
 - date and time scheduling 3-3
 - overview 3-6
- Scheduling
 - entering parameters 8-2
 - JCL overrides 10-6
 - job, trigger 3-43
 - Menu 3-9
 - overview 3-2
 - Relative days 3-19, 3-32
 - screen 3-9
- SCHID field 3-15, 3-28, 3-40, 3-48, 3-53
- Screen to screen transfer 1-20
- Screens
 - Active Job Resources Display 5-15
 - Application System Documentation 7-23
 - bypassing 1-18
 - common fields 1-15
 - Corequisite Resources list 5-23
 - CPU Job
 - Definition 2-3
 - Documentation 7-4
 - Predecessors 4-13
 - Scheduling 3-11
 - Scheduling Parameter Edit 3-14
 - Data Set
 - Definition 9-5
 - Documentation 7-15
 - Predecessors 4-8
 - Triggering 3-46
 - DB Menu 1-17

Screens (*continued*)

- DD Statement Documentation 7-19
- edit 3-12, 3-39
- Function Menu 1-16
- Input Network
 - Scheduling 3-24
 - Scheduling Parameter Edit 3-27
 - Triggering 3-45
- Input/Output Network
 - Definition 8-3
 - Documentation 7-7
 - Tasks 4-19
- JCL Library Maintenance 10-2
- Job
 - Predecessor/Successor Menu 4-6
 - Resource Management 5-6
 - Triggering 3-44
- Job/resource Cross Reference List 5-12
- Jobs Waiting on Resources 5-20
- layout 1-15
- Modification to Resolved Schedule Dates 3-52
- Output Network
 - Scheduling 3-36
 - Scheduling Parameter Edit 3-39
- Pending Resources Job Display 5-18
- Report IDs Created 4-29
- Resource Count Resource Management 5-25
- RM menu 5-5
- Scheduling Menu 3-9
- User Memo-Form Predecessors 4-25
- User-Defined Item Documentation 7-11
- Workload Documentation Menu 7-3

Security 1-10, 13-7

- overview 1-10
- user ID 13-7

Selecting

- Database Maintenance function 1-17
- documentation functions 7-3

Sending messages 10-16

Sequence numbers 13-7

Shared resources 5-2

Shortcuts

- bypass menus/screens 1-18, 1-22
- LIST function 1-19

Special

- documentation
 - features 7-26
 - messages 8-5
- override library 2-6, 10-29
- purpose documentation 7-33

SR (SAVE/REPL) function

- input network schedule 3-28
- job schedule 3-15
- output network schedule 3-40

SS (SAVE/SAVE) function

- SAVE input network schedule in database 3-28
- SAVE job schedule in database 3-15
- SAVE output network schedule in database 3-40

STATUS function, alias 1-20

Status of modifications made to schedule 3-53

Submitting

- JCL in text editor Active Area to default MAINID 10-3
- time of day
 - for schedule ID 3-17
 - requirement on TRGD-JOB 3-49

SUBTM command 4-5

Summary of release enhancements 1-1

SYSIN control cards 13-55

System level documentation screen 7-23

System Management Facility (SMF) interface 9-7

T

Task completed 4-25

Temporary JCL 10-29, 10-31

Temporary predecessors 4-4

Text editor 1-13, 11-1

Timing

- for generation of base calendar 3-3

- permanent discontinuation of job 2-11

- start of new jobs 2-11

Top line commands 1-14

Transferring

- from screen to screen 1-20

- to another DB function (batch input) 1-23

- to CPU Job Scheduling Parameter Edit screen 3-12

- to Input Network Scheduling Parameter Edit screen 3-25
- to Output Network Scheduling Parameter Edit screen 3-37

- work from one database to another 13-1

TRIG command 12-6

Trigger scheduling of job 3-43

Triggers 3-6, 13-37

TSO/ISPF interface 11-30

TYPE field 9-7

U

UCC7Rxxx, job name for workload balancing 2-5

Unconnected workstation networks 13-21

Unsatisfied requirements 4-5

- UPD function, alias 1-20
- Updating
 - base calendars using DB.2.8 screen 3-55
 - data set
 - characteristics 9-10
 - information in database 9-6
 - requirements 4-9
 - DESC and LINK fields of
 - job level documentation member 7-5
 - system level documentation member 7-24
 - formatted screen fields of
 - DB.4.2 screen 7-8
 - DB.4.3 screen 7-12
 - DB.4.4 screen 7-16
 - DB.4.5 screen 7-20
 - input network SCHID-COUNT and SCHID in work area 3-28
 - job
 - characteristics 2-16
 - information in database 2-5
 - SCHID-COUNT and SCHID in work area 3-15
 - network
 - information in database 8-4
 - requirements 4-20
 - output network SCHID-COUNT and SCHID in work area 3-40
 - predecessor job requirements in database 4-14
 - schedule information in database for job or network 3-53
 - to remove network station 8-5
 - triggering element 3-47
 - user requirement 4-26
- UPPER FEM subcommand 11-18
- User
 - ID security 13-7
 - level documentation 7-11, 13-6
 - Memo-Form Predecessors screen 4-25
 - requirement 4-25
- User-defined
 - documentation segments 7-26
 - Item Documentation screen 7-11
- Using temporary JCL 10-30, 10-31
- USR connections 13-16
- Utility commands 1-10
 - RENAME 9-6, 9-10

V

- Variable parameter in ARF 6-38
- VER statement 10-13
- VERIFY command 2-7, 4-5

- Verifying logical pointers in database 12-2
- VRM (Virtual Resource Management) 1-8
 - and database transportability 13-53
 - DBT Job Extract JCL 13-54
 - device control 5-28
 - Job Extract Report 13-53
 - Menu screen 5-5
 - overview 5-1

W

- WLB
 - job class 2-13
 - job name, UCC7Rxxx 2-5
 - job priority 2-14
- Work flow control 1-8
 - commands
 - DEMAND 2-2, 3-7, 4-12, 4-24
 - DEMANDH 3-7
 - DMDNW 3-8, 8-2
 - JCLOVRD 2-7
 - LOAD 2-2, 2-16, 4-12, 4-24, 10-28
 - LOADH 2-13, 10-28
 - POST 4-5, 4-25
 - RUN 2-2, 3-8, 4-12, 4-24
 - RUNH 2-13, 3-8
 - RUNNW 3-8, 8-2
 - SUBTM 4-5
 - VERIFY 2-7, 4-5
- Work scheduling 3-2
- Workload
 - balancing
 - overview 1-12
 - documentation 1-12, 7-1
 - documentation menu screen 7-3
 - forecasting 1-11
 - planning 1-12
 - scheduling 1-8
 - sequencing 1-8
- Workstation
 - network control
 - See* Work flow control
 - networks 8-1, 13-6, 13-19

X

- XEND statement 10-10
- XI statement 10-10
- XO statement 10-10
- XREF command 12-6

Y

YEAR field 3-54

