CA Workload Automation SE

Database Maintenance Guide

r11.3

Third Edition

This Documentation, which includes embedded help systems and electronically distributed materials (hereinafter referred to as the “Documentation”), is for your informational purposes only and is subject to change or withdrawal by CA at any time. This Documentation is proprietary information of CA and may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of CA.

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

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

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

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

The manufacturer of this Documentation is CA.

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

Copyright © 2015 CA. All rights reserved. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

CA Technologies Product References

This document references the following CA products:

- CA Workload Automation SE, formerly CA 7® Workload Automation
- CA ACF2™
- CA Dispatch™
- CA Easytrieve® Report Generator (CA Easytrieve)
- CA JCLCheck™ Workload Automation (CA JCLCheck WA)
- CA Librarian®
- CA Panvalet® for z/OS
- CA Roscoe® Interactive Environment (CA Roscoe IE)
- CA Top Secret®
- CA Workload Automation AE
- CA Workload Automation iDash
- CA Workload Automation Restart Option for z/OS Schedulers (CA WA Restart Option), formerly CA 11™ Workload Automation Restart and Tracking

Contact CA Technologies

Contact CA Support

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

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

Providing Feedback About Product Documentation

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

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

Documentation Changes

The following documentation updates have been made since the last release of this documentation:

- Monitoring Service Level Agreements (see page 26)—Topic includes information about CA -7 Server for iDash.
- Workload Monitoring (see page 27)—Topic includes information about CA -7 Server for iDash.

Contents

Chapter 1: Introduction 17

Overview 17
Functional Overview 17
- Online Assistance 17
- Workload Scheduling 18
- Workload Sequencing 18
- Work Flow Control 19
- Virtual Resource Management 19
- Interface with IBM Workload Manager (WLM) 19
- Automated Recovery Facility 20
- Job Restart 21
- Online Utility Execution 21
- Security 21
- JCL Overrides 22
- External Communications Facilities 22
- Batch Terminal Interfaces 23
- Batch Card Load Program 23
- Workload Forecasting 23
- Workload Planning 24
- Workload Balancing 24
- Workload Documentation 24
- Management Level Reporting 24
- Text Editor 25
- Event Notifications 26
- Ensuring Availability 26
- Monitoring Service Level Agreements 26
- Workload Monitoring 27
- Disaster Recovery Mode 27
- Database Transportability 27
- Network Communications Facility 27
- Controlling the Enterprise Through CA WCC 28
- Interface with z/OS UNIX System Services 28

Online Input 28
- PF Keys 29
- Formatted Panel Messages 29
- CA-7 Function Menu 30
- DB Menu Panel 31

Bypassing the Menu 31
Function Shortcuts 33
Function Transfer 33
Aliases for Formatted Panel Functions 34
Batch Input 35
Online Format 36
Batch Format 36
Function Shortcuts and Bypassing the Menu 36
Function Transfer 37
Aliases for Formatted Panel Functions 37

Chapter 2: Setting Up and Maintaining Jobs 39

Types of Jobs 39
How You Add a Job 40
DB.1 - CPU Job Definition Panel 41
Usage Notes 56
Example: Batch Input 56
How You Change a CPU Job 57
DB.10 - XP Job Definition Panel 57
Usage Notes 71
Example: Batch Input 71
DB.11 - Agent Job Definition Panel 71
Usage Notes 82
Example: Batch Input 82

Chapter 3: Scheduling Work 83

Work Scheduling 83
Date and Time Scheduling 84
Schedule Modifications 87
Event Scheduling with Triggers 88
On-Request Scheduling 89
Schedule Definition 91
DB.2 - Scheduling Menu Panel 91
DB.2.1 - Job Scheduling Panel 92
Job Scheduling Parameter Edit Panel 95
DB.2.2 - Input Network Scheduling Panel 108
Input Network Scheduling Parameter Edit Panel 111
DB.2.3 - Output Network Scheduling Panel 120
Output Network Scheduling Parameter Edit Panel 123
Trigger Scheduling Panels 128

DB.2.4 - Job Triggering Panel 128
DB.2.5 - Input Network Triggering Panel 130
DB.2.6 - Data Set Triggering Panel 131
DB.2.7 - Modification to Resolved Schedule Dates Panel 137
DB.2.8 - Base Calendar Maintenance Panel 140

Chapter 4: Defining Requirements 145

Execution Requirements and the LOAD Process 145
Dependence Definition 145
Database Definitions 146
Data Set Dependencies 146
Other Dependencies 147
Temporary Predecessors 147
Satisfy Requirements 148
Effect of Time Zone Normalization 149
Predecessor and Successor Definition 149
DB.3 - Job Predecessor/Successor Menu Panel 150
DB.3.1 - Data Set Predecessors Panel 151
DB.3.2 - Job Predecessors Panel 155
DB.3.4 - Input/Output Network Tasks Panel 160
DB.3.6 - User Memo-Form Predecessors Panel 166
DB.3.7 - Report IDs Created Panel 170

Chapter 5: Virtual Resource Management 173

Resource Types 173
Shared Resources 174
Exclusive Resource 174
Corequisite Resources 174
Address Space Resources 174
Resource Count Resources 175
VRM Variable Definitions 175
Resource Conflicts Resolution 179
RM - Virtual Resource Management Menu Panel 179
RM.1 - Job Resource Management Panel 180
Example: Batch Input 186
Critical Path Monitoring Corequisite Resources 186
RM.2 - Job/Resource Cross-Reference List Panel 187
Example: Batch Input 190
RM.3 - Active Job Resources Display Panel 190
Example: Batch Input 193

RM.4 - Pending Resources Job Display Panel 193
Example: Batch Input 195
RM.5 - Jobs Waiting on Resources Panel 196
Example: Batch Input 198
RM.6 - Corequisite Resources List Panel 198
Example: Batch Input 200
RM.7 - Resource Count Resource Management Panel 200
Example: Batch Input 203
VRM Device Control 203
VRM Device Control Functions - Definition Structure 204
VRM Device Control Functions - Definition Control 205
VRM Device Control Functions - Submission Control 207
VRM Device Control - Activation 207

Chapter 6: Automated Recovery Facility 209

Overview 209
Monitor Exception Conditions 210
Respond to Exception Conditions 211
Types of Exceptions Recognized by ARF 211
ARFSET Structure 214
ARF Definition Structure 215
Filter Criteria 215
Type Specific Tests 215
Responses 216
Final Disposition 217
Implementation Considerations 217
AR.3 - ARF Condition Definition Maintenance Panel 218
ARF Condition Definition Edit Panel 220
Field Descriptions 221
Field Descriptions - Filter Criteria 223
Field Descriptions - Type Specific Tests for JC and SC Conditions 228
Field Descriptions - Type Specific Tests for EC, EE, IS, LB, LE and LS Conditions 233
Field Descriptions - Responses 237
Field Descriptions - Final Disposition 238
Rules for Coding ARF Action Statements 241
AC - Issue a Command 242
AM - Issue a Message 243
AW - Wait 244
AJ - Schedule a Recovery Job 245
Use Reserved Words in Type Specific Tests 246
Use Variables in ARF Action Statements 248

Examples of ARF Condition Definition....249
Defining a Job Completion Condition (JC)....249
Defining a Late at Job End Notification (LE)....251

Chapter 7: Workload Documentation 253

Overview....253
Workload Documentation Panels....254
DB.4 - Workload Documentation Menu Panel....254
DB.4.1 - CPU Job Documentation Panel....255
DB.4.2 - Input/Output Network Documentation Panel....258
DB.4.3 - User-Defined Item Documentation Panel....262
DB.4.4 - Data Set Documentation Panel....266
DB.4.5 - DD Statement Documentation Panel....270
DB.4.6 - Application System Documentation Panel....274
Special Documentation Features....277
User-Defined Documentation Segments....278
Special Purpose Documentation....285
#QDESC Segment Usage....286
#station-name Segment Usage....287
#RESTART Segment Usage....288
#END Statement....288

Chapter 8: Workstation Networks 291

Network Adds....291
DB.5 - Input/Output Network Definition Panel....292
Example: Batch Input....295
Network Changes....295

Chapter 9: Data Sets 297

Data Set Adds....297
Permanent Data Sets....298
Dynamically Allocated Data Sets....298
External Data Sets....299
DB.6 - Data Set Definition Panel....299
Usage Notes....304
Example: Batch Input....304
Data Set Changes....304

Chapter 10: JCL and PARM Data Maintenance 305

Parameter Data Maintenance for Internal Cross-Platform Job Types 305
DB.7 - JCL Library Maintenance Panel 306
Usage Notes 311
Scheduled Overrides 311
#JI and #JO Statements 312
#JEND Statement 315
#XI, #XO, and #XEND Statements 316
Scheduled Override Examples 316
Additional Override Statements 318
#ARF Statement 319
#MSG Statement 320
#RES Statement 321
#SCC Statement 322
Additional Override Examples 325
SASSJCLU - JCL Utility 328
Usage Notes 329
Control Statement 329
JCL 330
JCL Validation 330
LOAD Command Processing 331
Special Override Library 331
Special Override Library Definition 332
Temporary JCL Creation 332
Temporary JCL Usage 332
Other Considerations 333
Alternate JCL Libraries 333
Alternate Library Definition 333
Temporary JCL Creation 333
Temporary JCL Usage 334
Other Considerations 334

Chapter 11: Using the Edit Facility 335

Overview 335
Active Area 336
Edit Work File 337
Environments 337
Editor Usage 337
Invoking the Editor 338
Editing Text Data 338
Leaving the Editor 338

CA 7 Text Editor Environment 338
Edit Modes 339
Updating Text 359
Creating Text 359
Special Considerations 360
TSO/ISPF Editor Environment 361
Requesting Edit Functions from the ISPF Editor 362
Default Requests for CA Workload Automation SE Edit Functions 363
Special Considerations 364

Chapter 12: Internal Cross-Platform 367

XN.1 - XNODE Command 367
XN.2 - XPSWD Command 370
(Password Present)/(Password Not Present) Message 374
XN.3 - AGPSWD Command 375

Chapter 13: Database Verification 379

Pointer Verification 379
Pointer Verification Parameters 380
Pointer Verification by Record Type 381
Analyze Commands 384

Chapter 14: Database Transportability 387

Overview 387
Assumptions 389
LOAD/RELOAD Status 389
CA WA Restart Option Step Insertion 390
Base Calendars 390
Calendar Schedules 390
DB.2.7 Schedule Modifications 390
NXTCYC Settings 391
Execution JCL Libraries 391
Cataloged Procedures 392
In-stream JCL Procedures 392
Workstation Networks 392
User Level Documentation 392
XPJOB/AGJOB Considerations 393
ANALYZE Application 393
Commas in the Data 393
Sequence Numbers 394

User ID Security 394
LINKed Documentation 394
Database Extraction 395
Moving Jobs 399
Add at New Site 400
Requirements and Network Connections 402
Delete at Original Site 404
Disable at Original Site 405
Workstation Networks 406
Add at New Site 407
Delete at Original Site 407
Disable at Original Site 407
Unconnected Workstation Networks 408
Application Data Sets 408
Add at New Site 409
Delete at Original Site 410
PDS JCL Members 410
Add at New Site 411
Delete at Original Site 412
CA Librarian JCL 412
Add at New Site 413
Delete at Original Site 414
CA Panvalet JCL 415
Add at New Site 416
Disable at Original Site 417
Cataloged Procedures 418
Add at New Site 419
Delete at Original Site 419
Documentation 420
Add at New Site 421
Delete at Original Site 422
Schedules 422
Add at New Site 422
Delete at Original Site 426
Node and Access File 427
Agent DIV File 428
Create Command Files 429
Job 1 429
Job 2 432
Job 3 433
Special Considerations 438
Design Limitations 438

Add to the New Database 439
Run at New Site 440
Mass Changes at Existing Site 440
Virtual Resource Management Database Extracts 440
VRM DBT Program 441
VRM DBT Job Extract JCL 442
VRM Node and Access Transport Program 445
VRM Node and Access Extract JCL 446
PARM Keywords 447
Data Sets Used/Created 447
Output Reports 448
Automated Recovery Facility Database Extracts 449
ARF DBT Extract JCL 450
Data Sets Used/Created 452

Chapter 15: Mass Job Rename Utility 453

Overview 453
Mass Job Rename Procedure 454
Mass Job Rename Control File (RENAMES) 456
Database Rename Process (CAL2JRXD) 456
CAL2JRXD Process JCL 457
CAL2JRXD Sample Reports 458
CAL2JRXD Specific Element Changes 459
Error Conditions 459
Queue Rename Process (CAL2JRXQ) 460
CAL2JRXQ Process JCL 461
CAL2JRXQ Sample Reports 462
Specific Element Changes 463
Error Conditions 463
VRM Process (CAL2JRXV) 463
CAL2JRXV Process JCL 464
CAL2JRXV Sample Reports 465
Specific Element Changes 466
Error Conditions 467
ARF Process (CAL2JRXA) 467
CAL2JRXA Process JCL 468
Specific Element Changes 470
Error Conditions 471

Appendix A: Using CA 7/PERSONAL SCHEDULING 473

Overview 473
Panel Descriptions 474
Access CA-7/PERSONAL SCHEDULING 477
Add a Job 479
Submit a Job 480
Display a Job's Status 481
Implement CA 7/PS 482
    Defining Additional PATTERN Jobs 484
CA 7/PS Messages 484

Index 493

Chapter 1: Introduction

The Database Maintenance Guide is intended for database administrators, change control personnel, or both using CA Workload Automation SE.

This section contains the following topics:

Overview (see page 17)
Functional Overview (see page 17)
Online Input (see page 28)
Batch Input (see page 35)

Overview

CA Workload Automation SE 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 Workload Automation SE is an online, realtime, interactive system that automatically controls, schedules, and initiates work according to time-driven and/or event-driven activities.

Functional Overview

CA Workload Automation SE capabilities and facilities include, but are not limited to, the following topics.

Online Assistance

CA Workload Automation SE incorporates facilities to assist in using the system to control the production environment, to operate the system, and to retrieve information from the system.

HELP Facility

The HELP facility is an online function available at any CA Workload Automation SE terminal. Information describing 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 explanation of the requested application or command. Information concerning commands includes a brief description of the command, its keywords, formats, and uses.

PF Key Assignment

Program function (PF) keys can 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.

Workload Scheduling

CA Workload Automation SE 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 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 job processing.

CA Workload Automation SE also has the ability to track batch jobs and started tasks that are submitted outside of CA Workload Automation SE 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 that are to be tracked. This is only available for CPUs that share the Communication data set with CA Workload Automation SE.

CA Workload Automation SE can also schedule workload to systems outside the CA Workload Automation SE system using cross-platform scheduling. These jobs execute on a distributed system, and the completion data is routed back to CA Workload Automation SE so that subsequent work can be brought in. The connections can be either through a CAICCI network or through a TCP/IP network.

Workload Sequencing

Workload sequencing refers to facilities that 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.

Work Flow Control

Although CA Workload Automation SE automatically schedules and invokes a defined workload, sometimes it is necessary to circumvent scheduled work flow for new priorities. With CA Workload Automation SE, 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 that might otherwise be overlooked or delayed.

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 Workload Automation SE during the job submission and job/step completion processes and can be used to serialize a production cycle or critical resource access.

Interface with IBM Workload Manager (WLM)

If specified, CA Workload Automation SE can set the IBM WLM resource state to on when using it as system initialization starts. When the address space is shut down, it sets the resource state to off. Also, CA Workload Automation SE can insert the designated scheduling environment (SCHENV=) parameter into the JCL it submits, based on a default or a name specified through a VRM resource definition. This lets you base the routing of the job execution on the availability of WLM resources.

Automated Recovery Facility

The Automated Recovery Facility (ARF) for CA Workload Automation SE 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 can 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 job definition panel for the job.

ARF specifically tests for the exception conditions that are described in the ARFSET for the job. Each ARFSET can 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 can specify criteria that are precisely tailored to fit exact exception conditions for an individual job or can 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 Workload Automation SE or MVS commands. ARF can also submit and track special recovery jobs that are not part of the "normal" schedule. Special messages can 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 Workload Automation SE or MVS commands immediately at the point in time when ARF detects a step completion code exception.

Job Restart

Under CA Workload Automation SE, jobs that 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 can 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.

Internal cross-platform jobs, known as XPJOBs or agent jobs, contain only one step: the information sent to the destination. These job types are restartable only from the beginning and cannot use CA WA Restart Option facilities if available.

If CA WA Restart Option is available, both restart cleanup and job restart can be accomplished online on a single screen. CA Workload Automation SE can tell CA WA Restart Option which step to restart, based on condition codes interpreted by CA Workload Automation SE.

Online Utility Execution

The CA Workload Automation SE 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.

The online execution of utilities reduces time that is otherwise required to perform these functions. The online execution also eliminates the dependence on other methods like batch jobs, TSO, CA Roscoe, and so forth.

Security

CA Workload Automation SE 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 Workload Automation SE, 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 Workload Automation SE.

An internal security matrix can be used if an external security package is not available or desired. With the CA Workload Automation SE security matrix, the following 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, exits are available that allow interfacing with external security packages.

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 Workload Automation SE 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

Also, the global variable feature allows for variable substitution in JCL either with CA Driver or without using CA Driver.

External Communications Facilities

CA Workload Automation SE provides several programs that facilitate communications between CA Workload Automation SE and users outside the CA Workload Automation SE 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 Workload Automation SE about the availability of processing requirements for CA Workload Automation SE jobs.

Batch Terminal Interfaces

CA Workload Automation SE provides various terminal interfaces to communicate with and receive output from the online environment. The interfaces support most commands.

- The Batch Terminal Interface (BTI) uses the communication data set to pass commands to the online environment for execution.
- The CAICCI terminal interface uses the CAI Common Communications Interface network to pass commands to the online environment. The interface can be used on any system that connects through the CAICCI network to the destination CA Workload Automation SE environment.
- The TCP/IP terminal interface uses the TCP/IP sockets to communicate with the desired online environment from sites that are connected using TCP/IP.

Batch Card Load Program

The Batch Card Load Program (BCLP) loads card-image data into data sets that can be specified as input requirements for CA Workload Automation SE 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.

Workload Forecasting

The Workload Forecast facility provides several important functions. The Workload Forecast facility lets you:

- Project the scheduled workload for a given period.
- 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.

In addition to the online forecasting facility, the Jobflow Illustrator forecasting facility can execute in a batch mode outside of the online environment and provides several features that the online forecasting tool does not provide:

- In addition to triggers, the forecasted flow reflects job dependencies such as other jobs and data sets.
- Nice visuals are available through the Microsoft Visio™ interface.
- Permits job additions or job deletions to see what effect occurs on the workload.

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. Since the CA Workload Automation SE normal production control and scheduling functions are not affected by the planning function, online processing continues as normal throughout the batch planning process.

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 Workload Automation SE. These objectives can be modified online at any time to accommodate situations that are not anticipated. CA Workload Automation SE automatically selects and submits a mix of jobs to the host system that best meets the defined objective.

For internal cross-platform jobs (XPJOBs or agent jobs), only WLB class and priority are considered. Because internal cross-platform jobs execute on other platforms, tape usage is not considered.

Workload Documentation

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

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

Management Level Reporting

Several tools are included that provide information about productivity of a data center. Management level reporting is done on a global basis for jobs and networks from historical data that the product retains. Four categories of historical reports are provided.

Automated Performance Analysis (APA) Reporting

APA can provide reports on performance information based on history. Information is available in 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.

History Reporting

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

CA Earl Reporting

An abbreviated yet powerful version of CA Earl is provided with CA Common Services. With CA Earl, several statistical type history analysis reports are provided. Some database inventory type reports are also provided.

CA Easytrieve Reporting

CA Easytrieve provides report definitions to produce the same statistical analysis and database inventory reports that CA Earl can produce. CA Common Services provides an abbreviated version of CA Easytrieve.

Text Editor

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

Event Notifications

CA Workload Automation SE interfaces with a variety of external products to notify users of events that have occurred. CA Workload Automation SE can pass job event information to the CA OPS/MVS EMA Application Programming Interface (API) upon which rules with intended actions are taken. This process uses the Master Station Message Routing (MSMR) feature, which can also be used with CA Generalized Transaction Server Event Console Handling Option (ECHO) feature to route messages to a central monitoring console. The /EMAIL command can issue emails, which can be coded in an ARF set to notify of job abends. Also, an interface to CA Service Desk can open issues based on selected events. Both the email and CA Service Desk interfaces support a variety of variables to customize the information communicated.

Ensuring Availability

CA Workload Automation SE can register with various interfaces to ensure it is available in the system. CA Workload Automation SE, during its initialization with the system occurring at CAIRIM, can register with the IBM Health Checker feature. This lets CA Workload Automation SE self-test itself to ensure that it is still active in the system and issue messages through Health Checker if the tests fail. When CA Workload Automation SE registers with the IBM Automatic Restart Option (ARM), and CA Workload Automation SE has ended unexpectedly in the system, ARM can restart CA Workload Automation SE or its ICOM based on the established ARM policy. With CA OPS/MVS EMA, CA Workload Automation SE can register with the System State Manager to inform its status (up, down, and so forth).

Monitoring Service Level Agreements

CA Workload Automation SE interfaces with CA Workload Automation iDash, through a separate address space called CA-7 Server for iDash, to provide a complete view of all defined and actively running workload within CA Workload Automation SE. CA Workload Automation iDash is the product that actually defines and tracks SLAs. You can define and view this information from within the CA Workload Automation iDash interface.

CA Workload Automation SE interfaces with CA Critical Path Monitor (CA CPM) by defining a flow of jobs and specifying a flow completion time. CA Workload Automation SE passes events to CA CPM regarding each job within that flow when the first job starts. CA CPM is the product that actually "predicts" when a flow may not complete by the specified time. You can view the time from the CA CPM ISPF interface.

Workload Monitoring

CA-7 Server for iDash, a separate address space from CA Workload Automation SE, can optionally be activated to monitor and stream the current workload being processed by CA Workload Automation SE. CA-7 Server for iDash provides a live stream of events as they occur within CA Workload Automation SE. CA-7 Server for iDash can provide this information to CA Workload Automation iDash.

Jobflow Monitor (JFM), a separate address space from CA Workload Automation SE, can optionally be activated to track and monitor the current and near future workload being processed by CA Workload Automation SE. JFM provides a complete picture of what CA Workload Automation SE is currently doing and needs to do for the next *n* hours (*n* is a user-defined number). JFM can provide this information to other interfaces, such as CA Critical Path Monitor.

Disaster Recovery Mode

When defining the workload to CA Workload Automation SE, users can assign a disaster recovery class to indicate the importance of executing the job should a disaster occur. When activating CA Workload Automation SE in a disaster situation, users can designate which classes must be executed, and which classes are less critical and need not be executed. The online workload forecasting tool can display the jobs that would execute in a disaster mode.

Database Transportability

In some locations, workload is moved from one LPAR to another based on resource usage or other considerations. Sometimes the new LPAR is not included in the CA Workload Automation SE scope and therefore the jobs must be moved to the new location. With database transportability (DBT), you can export a set of jobs from one CA Workload Automation SE environment and import the set into another environment. DBT accomplishes this through a series of batch jobs. DBT moves everything associated with the job: relationships, JCL, and requirements.

Network Communications Facility

The Network Communications Facility (NCF) of CA Workload Automation SE tracks job status information that CA Workload Automation SE submits regardless of the JES NJE node that executed the job. A job is submitted by CA Workload Automation SE at one location. The JES /*ROUTE XEQ statement tells JES to route the job to a node for execution. As the job executes at the JES node, NCF tracks the job status and routes the information back to CA Workload Automation SE so that the job status is reflected accurately and requirements can be posted and subsequent jobs triggered.

Controlling the Enterprise Through CA WCC

CA Workload Control Center (CA WCC) is a graphical user interface (GUI) through a web browser that permits a user to control the Workload Automation enterprise from a single station. This includes multiple CA Workload Automation SE environments as well as CA Workload Automation AE environments. Jobs can be defined or edited, and the active workload can be monitored, making it easier to interface with the enterprise workload.

Interface with z/OS UNIX System Services

CA Workload Automation SE provides a method to invoke select commands from the z/OS UNIX System Services (USS) environment. This invocation lets scripts and commands execute in that environment to start events on CA Workload Automation SE. The CA GTS feature USS File Watch (USSF) can monitor the creation or update of files in the USS environment. This monitoring can send a command to CA Workload Automation SE to cause an event to occur. Typical events are demanding a job or posting a requirement.

Online Input

Before a user performs database maintenance, the user must successfully log in to a CA Workload Automation SE terminal.

A series of formatted panels enable online input to the CA Workload Automation SE database. Each major Database Maintenance (DB) command has an associated input panel. Any of the following items can display input panels:

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

When the wanted command or menu panel option is entered, a data input panel 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 can enter any CA Workload Automation SE command on the top line of the current panel.

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

When updating text fields through formatted panels in online mode, enter a space as the first character and erase to end-of-field. When the update occurs, the field in the database is removed. When using topline commands, omitting the keyword lets you take the default for that command.

More information:

DB Menu Panel (see page 31)

PF Keys

After a function is selected on the menu and the function panel is displayed, program function key 3, PF3 is temporarily set to return to the DB MENU panel. In native CA Workload Automation SE VTAM mode, any value that was previously assigned to PF3, by either the user or CA Workload Automation SE, is temporarily ignored as long as the function panel is being used and reverts 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 Workload Automation SE 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.

Formatted Panel Messages

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

PROGRAM:

Identifies the last four characters of the module name that processed the data of this panel.

MSG-INDX:

Identifies the return message identifier.

panel id

Identifies the panel ID of this panel.

yy.ddd/hh:mm:ss

Identifies the date and time of this returned message.

MSGS:

Identifies the 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. The field does not appear on general inquiry output in batch mode.

MESSAGE:

Identifies up to two lines of message text. Messages are listed in the *Message Reference Guide* as a combination of the PROGRAM and MSG-INDX values. For example, SM11-00.

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
```

DB Menu Panel

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

```
---------------------  CA-7 DATA BASE MAINTENANCE MENU  ----------------------
FUNCTION ===>

DATA BASE DEFINITION FOR:
   1 - CPU JOB
       A - CROSS PLATFORM (XPS) JOB DEFINITION
   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: SDM0  MSG-INDX: nn  -- DB       --  yy.ddd / hh:mm:ss
MESSAGE:
```

To display, enter:

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

To exit, enter the name of an online panel as the FUNCTION value or move the cursor to the top line and enter a top line command if some other function is wanted.

Usage Notes

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

Bypassing the Menu

CA Workload Automation SE provides an extensive menu service for Database Maintenance. The menus are structured to help a novice user step through various panels. However, as proficiency increases, it can be desirable to bypass primary menus and proceed directly to the wanted panel.

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

Bypassing Secondary Menus

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

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

Entering 2.3 on the DB Menu panel indicates that you want function 2 (SCHEDULING) and that Scheduling Menu function 3 (OUTPUT NETWORK) is the function that you want to perform.

CA Workload Automation SE displays the DB.2.3 panel directly and bypasses the Scheduling Menu (DB.2).

You can also bypass these secondary menu panels. Enter the equivalent top line command, placing a comma after the command and selecting a category, as follows:

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

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

Bypassing Tertiary Menus

If your site has enabled CA Workload Automation SE agents, Function A has tertiary menus. To go directly from the DB Menu panel to a function that appears on one of the tertiary menus, enter FUNCTION in the *n.n.n* format. See the following example:

```
----------------------
FUNCTION ===> A.F.D
```

Entering A.F.D on the DB Menu panel indicates that you want function A (CROSS PLATFORM (XPS) JOB DEFINITION), then CA-7 Cross Platform Jobs Menu function F (SAP), and that CA-7 SAP Jobs Menu D (SAP Job) is the function to perform.

CA Workload Automation SE displays the Agent Job Definition panel directly, bypassing the CA-7 Cross Platform Jobs Menu and CA-7 SAP Jobs Menu.

The menus provide a view of the various agent job types that are available for scheduling under CA Workload Automation SE. Using the menus also pre-fills the Agent Job Type field on the CA-7 Agent Job Definition panel (DB.11).

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 wanted are included in the top line command following the category, as follows:

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

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

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

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

  This example lists all jobs and schedules that are triggered by the completion of input network PAYPUNCH.

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

Shortcut Examples

The following are more shortcut examples:

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

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

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

Function Transfer

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

Aliases for Formatted Panel Functions

All function values for formatted panels can be assigned alternate (alias) names. Aliases allow for user-designated abbreviations (for example, L for LIST). Aliases also permit alternate values such as CHANGE for UPD. The only restrictions are that the alias names must be eight characters or less, and must not conflict with a function or other alias names.

The following table is a list of formatted panel function values and some alias names that are distributed with the product:

Functions	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
PURGE	DELETE	N/A
QUIT	N/A	Q
REFRESH	UPDATE	REF
RENAME	UPDATE	REN
REPL	UPDATE	R,REP,REPLH
REQ	UPDATE	N/A

Functions	Service Level	Alias
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
STATUS	READ	ST
SUBMIT	SUBMIT	SUB
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

Note: For more information about modifying the alias values, see the *Systems Programming Guide*.

Batch Input

Two different formats for transaction input and output are supported.

Online Format

The Online format is used in CA Workload Automation SE online terminal sessions. Online terminal sessions support the command-line input and output and data transfer using formatted panels.

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

A command transaction begins with the command name and parameters can follow. These transactions are documented in the *Command Reference Guide*.

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

Batch Format

The external communicators like the Batch Terminal Interface (BTI) and the CAICCI Terminal Interface use the batch format.

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

In some cases, batch and online transaction formats differ. For those cases, the documentation of each online function includes the description of the corresponding batch transaction format.

Note: For more information about external communicators, see the *Interface Reference Guide*.

Function Shortcuts and Bypassing the Menu

The shortcuts available online to bypass certain panels are of limited value in batch. Batch transactions must single step from function to function much as a new user steps through a sequence of equivalent panels (for example, DBM, SCHD, JOB, and others). Some commands that provide secondary menu panels 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.

Function Transfer

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

Aliases for Formatted Panel Functions

Batch commands can use any of the alias names that were assigned just as they are used online.

Note: For the procedures for assigning alias names, see the "User Exits and Modifications" chapter of the *Systems Programming Guide*.

Chapter 2: Setting Up and Maintaining Jobs

This section describes setting up and maintaining a job in the CA Workload Automation SE database.

This section contains the following topics:

Types of Jobs (see page 39)
How You Add a Job (see page 40)
DB.1 - CPU Job Definition Panel (see page 41)
How You Change a CPU Job (see page 57)
DB.10 - XP Job Definition Panel (see page 57)
DB.11 - Agent Job Definition Panel (see page 71)

Types of Jobs

You can define and schedule up to three types of jobs through CA Workload Automation SE.

- The first job type is the CPU job. This type of job is submitted to the Job Entry System (JES) of a z/OS operating system.
- The second job type is the XPJOB. This type of job executes on another platform that is executing another CA scheduling system or a CA job management agent, such as CA UJMA.
- The third job type is actually a job category, CA Workload Automation SE agent jobs. These jobs can be FTP jobs, SAP jobs, database jobs, and more. CA Workload Automation SE submits these jobs to execute on a platform running the CA system agent and associated plug-in.

Unless stated, "job" applies to all job types. If the word job has an adjective of CPU, XPJOB, or agent, the sentence or section applies to the specific job type mentioned. If something applies only to XPJOB or an agent job, the words "internal cross-platform job" are used (this designation does not include CPU jobs).

When requesting information about a job, you can enter the request from the JOB (CPU), XPJOB, or Agent Job Definition panels. CA Workload Automation SE tests the type of job being requested and redirects processing to the appropriate job definition panel if necessary.

For example, if you request information about a CPU job from the XP Job Definition panel, you are redirected to the JOB Definition panel. Likewise, if you request information about an agent FTP job from the JOB Definition panel, you are redirected to the Agent Job Definition panel. This redirection applies to the LIST and any DELETE functions. If the function is UPDATE, you must be on the panel representing the job type. If an UPDATE function is attempted from the incorrect panel, an error message is issued.

How You Add a Job

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

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

  Note: For more information about BTI, see the *Interface Reference Guide*.

- If the JCL library is defined to CA Workload Automation SE but the jobs are not added to the database, you can use the DEMAND command to add the job to the database. This command causes the job to execute. The execution includes a LOAD step because the job is not defined in the database. You can also use the LOAD command 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 must be met.

- The job name of the job to be added must be the same as the JCL member name in the JCL library.
- The default values defined for the DB.1 panel must satisfy the job characteristics. The default values for the DB.1 panel can be overridden by defining a job that is named DEFAULTS with the values wanted. Also, if CLASS, JCLID, or JCLLIB is specified on the command, these values can override the DEFAULTS job values. System defaults cannot be overridden with the DEMAND and LOAD commands. Job characteristics, not covered by the defaults, must be manually checked.
- The JCL library ID number or symbolic JCL library name must be specified on the command. One JCL library should have an index value of 0 (zero). With the zero value, you do not have to enter the DB.1 panel JCLID field for jobs with JCL residing in that library when issuing the top line JCL command.

The DEMAND command represents on-request scheduling that is useful on a first-time basis. As an ongoing practice, we do not recommend this method except for jobs that must always be handled as on-request work.

You can add internal cross-platform jobs through the DB.10 panel and DB.A.*x* panels or through an external terminal interface such as Batch Terminal Interface (BTI) or CAICCI Terminal Interface (CTI) batch job. You can demand an internal cross-platform job to cause its execution, but the DEMAND command always searches the database for definition. The DEMAND command does not add any cross-platform jobs to the CA Workload Automation SE database.

More information:

Batch Input (see page 35)

DB.1 - CPU Job Definition Panel

The DB.1 - CPU Job Definition panel lets you enter or review data related to CPU jobs.

To display the panel, enter:

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

To exit the panel:

- Press PF3 to return to the DBM Menu panel except with the TSO-ISPF interface when the PF3 key is assigned to another function.
- Move the cursor to the top line of the panel and enter a top line command for another function.

```
------------------------  CA-7 CPU JOB DEFINITION  -------------------------
FUNCTION: xxxxxxxx  (ADD,DELETE,DD,PURGE,DELPRRN,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   DRCLASS: xxxxxxxx
              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:
```

This panel contains the following fields:

FUNCTION

Specifies the function to be performed. Value must be the name of some other panel 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 that 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 WA Restart Option interface is active, deletes the CA WA Restart Option 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 panel of all input data.

LIST

Lists a job and job-related information. In batch, a formatted panel is not listed. Only a found or not found message is returned.

UPD

Updates database information about a job.

Batch keyword: Positional parameter

Note: A default interpretation can be set for the DELETE function in the CA Workload Automation SE initialization file. For more information, see the OPTIONS statement in the chapter "Initialization" of the *Systems Programming 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. For information about bypassing these security checks, see the BYPSEC keyword in the chapter "Security Initialization Options" of the *Security Reference Guide*.

JOB

Defines the job name on which to perform the indicated function.

Limits: 1 to 8 alphanumeric characters

Batch keyword: Positional parameter

Note: Job name format of UCC7R*xxx* (where *xxx* is any three 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 five characters.

This job name always overlays the job name on the JOB statement in the execution JCL.

GENERAL

Indicates that this section of the panel contains general information about the job. No input is permitted for this field.

SYSTEM

(Optional) Specifies the user-defined application system name of which this job is a part.

Limits: 1 to 8 alphanumeric characters. This field cannot contain a comma.

Batch keyword: SYSTEM

JOBNET

(Optional) Specifies the name of a CPU job network of which this job is a part.

Limits: 1 to 8 alphanumeric characters. This field cannot contain a comma.

Batch keyword: JOBNET

OWNER

(Optional) Specifies the ID identifying ownership of this job. Depending on the CA Workload Automation SE startup options taken, the external security product being used and contents of the JCL, this value can be offered to the external security package by CA Workload Automation SE at job submission time as the authority for this job to be executed.

Limits: 1 to 8 alphanumeric characters. 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.

Batch keyword: OWNER

UID

(Optional) Specifies the CA Workload Automation SE user security identification.

Default: 0 (no internal security protection)

Limits: 1 to 3 numeric characters from 0 to 255

Batch keyword: USERID

JCL

Indicates that this line of the panel contains JCL information about the job. No input is permitted for this field.

ID

(Optional) Specifies 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.

Default: 0

Limits: 1 to 3 numeric characters from 0 to 253

Batch keyword: JCLID

MEMBER

(Optional) Specifies the JCL library member name and required if member name and job name are different.

Default: Job name

Limits: 1 to 8 alphanumeric characters

Batch keyword: JCLMBR

RELOAD

(Optional) Specifies whether to reload this job's JCL (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, the job is not flagged for load processing unless the LOAD command is used. If RELOAD=Y, the job is flagged for load processing. If RELOAD=N, the job is not flagged for load processing unless it is the first time it has run in CA Workload Automation SE. 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.

Default: N

Batch keyword: RELOAD

EXEC

(Optional) Indicates whether to execute this job.

Y

Specifies to execute the job. This is the default.

N

Specifies that the job does not run but shows a normal completion as if it did run. JCL is not required for nonexecutable jobs.

Note: Consider the following items:

- Since nonexecutable jobs do not get executed, no data sets are created. This means that any jobs waiting for data set requirement that may have been created for the job execution are not created and thus jobs may wait on these data set requirements.
- VRM definitions are ignored for the nonexecutable jobs.
- Critical Path Management flows cannot be started with nonexecutable job and thus the critical path may not be monitored.
- ARF recovery is not invoked for nonexecutable jobs.

Default: Y

Batch keyword: EXEC

RETAIN-JCL

(Optional) Specifies whether to retain execution JCL in the trailer queue after a successful run (Y or N).

Default: N

Batch keyword: RETJCL

LIB

(Optional) Specifies a 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.

Limits: 2 to 16 alphanumeric characters beginning with ampersand (&)

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

Specifies that this section of the panel contains requirements information about the job. No input is permitted for this field.

HOLD

(Optional) Specifies whether to place this job in a hold status when it enters the request queue (Y or N).

Default: N

Batch keyword: HOLD

JCL-OVRD

(Optional) Specifies whether this job needs manual JCL overrides before it can be submitted (Y or N). Similar to the JCLOVRD command.

Default: N

Batch keyword: JCLOVR

USE-OVRD-LIB

(Optional) Specifies whether to retrieve the JCL 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

Batch keyword: JOVRDLIB

VERIFY

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

Default: N

Batch keyword: VERIFY

MAINT

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

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

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

Default: N

Batch keyword: MAINT

SATISFACTION LEAD-TIME

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

JOB

(Optional) Specifies the number of hours to be considered when satisfying job dependent requirements.

The following are the possible values:

0

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

99

Specifies 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, specifies that each predecessor job must have run within the last *nn* hours. Values for *nn* can be from 1 to 98.

Default: 0

Batch keyword: JOBLDTM

Note: JOB and DSN are on a global level for all job and data set requirements that have a LEADTM value of 0000 on the DB.3 panels. This applies to all SCHIDs that the job runs under.

DSN

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

Default: 0 (specifies to ignore this field)

Limits: 1 to 2 numeric characters from 0 to 99

Batch keyword: DSNLDTM

ARFSET

(Optional) Names the collection of ARF definitions that apply to this job. Remember that ARF recovery is not invoked for nonexecutable jobs.

Limits: 1 to 8 alphanumeric characters

Batch keyword: ARFSET

Note: The specified ARFSET must be defined in the ARF database.

EXECUTION

Indicates this panel section contains execution information about the job. No input is permitted for this field.

MAINID

(Optional) Specifies on which CPU the job can or cannot be scheduled. If the job requirements impose CPU restrictions, specify SY*n* or /SY*n* where *n* is the system number and / indicates not this system. System numbers must be consistent with the initialization file CPU statement MAINIDS parameters.

Default: ALL (lets job run on any CPU)

Limits: 1 to 4 alphanumeric characters

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

(Optional) Specifies whether to insert the CA WA Restart Option RMS step automatically at execution time by CA Workload Automation SE. Specify Y to insert the step with the CA WA Restart Option 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 WA Restart Option Option Table is inserted.

Note: For more information, see the *Interface Reference Guide*.

Default: N

Batch keyword: INSRTRMS or RESTART

COND-CODE

(Optional) Used with RO (relational operator) to define the job-level condition codes used to determine whether a job executes successfully.

Default: 0

Limits: 1 to 4 numeric characters from 0 to 4095

Batch keyword: CONDCODE

RO

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

The following are the possible values:

EQ

Specifies equal to.

LT

Specifies less than.

GT

Specifies greater than.

GE

Specifies greater than or equal to.

LE

Specifies less than or equal to.

NE

Specifies not equal to.

#S

Specifies to make step condition code tests based on #SCC statements in the JCL.

IG

Specifies to do no evaluation of the job. CA Workload Automation SE always assumes the job completes successfully, regardless of condition codes, abend codes, or runtime JCL errors. When this is used, the INSERT-RMS fields should be N.

0

Specifies to make no condition test.

Default: 0

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 internal use by CA Workload Automation SE only. It simply tells CA Workload Automation SE what action to take *after* the job completes. CA Workload Automation SE does not determine or control which steps are to be executed.

DONT SCHEDULE -- BEFORE

(Optional) Specifies not to schedule this job before this date and time. This field is especially helpful for timing the start of new jobs.

The following is a sample of panel input:

BEFORE: *yyddd hhmm*

yyddd

Defines the Julian date.

hhmm

Defines the time of day.

The following is a sample of batch input:

BDATE=*yyddd*,BTIME=*hhmm*

Default: All zeros

Limits: Julian date specified as *yyddd* and time specified as *hhmm*

Batch keyword: BDATE and BTIME

AFTER

(Optional) Specifies not to schedule this job after this date and time. This field is especially helpful for timing permanent discontinuation of a job.

The following is a sample of panel input:

AFTER: *yyddd hhmm*

yyddd

Defines the Julian date.

hhmm

Defines the time of day.

The following is a sample of batch input:

ADATE=*yyddd*,ATIME=*hhmm*

Default: 99999 0000

Limits: Julian date specified as *yyddd* and time specified as *hhmm*

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=09031 and BTIME=2359 and ADATE=09001 and ATIME=0000, the job does not schedule during the time from Jan. 1, 2009 through Jan. 31, 2009, but does schedule at all other times.

CA Workload Automation SE uses current date and time for comparisons.

MESSAGES

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

LTERM

(Optional) Specifies that messages about this job are to be routed to this logical terminal name.

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.

Limits: 1 to 8 alphanumeric characters

Batch keyword: LTERM

REQUIREMENT - LIST

(Optional) Specifies whether to list preexecution requirements for this job when it enters the request queue (Y or N).

Default: Y

Batch keyword: RQLST

PROMPTS

(Optional) Specifies whether to issue prompt messages if this job is late (Y or N).

Default: Y

Batch keyword: PROMPT

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

ERROR MSGS -- RQMTS NOT USED

(Optional) Specifies whether to issue error messages for job requirements not used (Y or N).

Default: Y

Batch keyword: RQMSG

DSN NOT FOUND

(Optional) Specifies whether to list error messages for data sets used at execution time but not found in the CA Workload Automation SE database (Y or N).

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

Batch keyword: DSNMSG

RESOURCES

Indicates resource information about the job. No input is permitted in this field.

REGION

(Optional) Specifies the region size required by this job (information only).

Default: 0

Limits: 1 to 4 numeric characters

Batch keyword: REGION

CLOCK-TIME

(Optional) CA Workload Automation SE 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.

Default:

0000

Limits: 4 numeric characters specified as *hhmm*, where *hh* can be 00 through 23 and *mm* can be from 00 to 59

Batch keyword: ELAPTM

CPU-TIME

(Optional) CA Workload Automation SE maintains weighted average CPU time usage values for each job in its database. (See the preceding note under CLOCK-TIME.)

Limits: 5 numeric characters specified as *mmmss*, where *mmm* is minutes and *ss* is seconds

Batch keyword: CPUTM

CLASS

(Optional) Specifies the CA Workload Automation SE 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. This value does not have to match the JOB statement CLASS value.

Default: A space

Limits: 1 alphanumeric character

Batch keyword: CLASS

PRTY

(Optional) Specifies the CA Workload Automation SE 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.

Default: 0

Limits: 1 to 3 numeric characters from 0 to 255

Batch keyword: PRTY

MSGCLASS

(Optional) Specifies the job's message class. This field is informational only. Even though this field can be changed, it does not cause the JCL to be changed. Also, no validity checking is done on this field.

Default: A space

Limits: 1 alphanumeric character

Batch keyword: MSGCLASS

DRCLASS

(Optional) Specifies the job's disaster recovery class. This field has no impact on processing during normal execution. When running in disaster recovery mode, the disaster recovery class is used to determine whether the job should execute.

Note: For more information, see disaster recovery mode in the *Systems Programming Guide*.

Limits: 1 to 8 alphanumeric characters. Disaster recovery class values must start with a letter, #, or $ (not @) and can include letters, #, $, @, and numbers. Disaster recovery classes cannot contain embedded blanks.

Batch Keyword: DRCLASS

TAPE DRIVES

Indicates that this line of the panel contains information about tape drives needed for the job. (If not using Workload Balancing, these fields are informational only.) No input is permitted for this field.

TYPE1

Indicates that the two following fields (M and C) contain information about TYPE1 tape drives need for the job. No input is allowed for this field.

M

(Optional) Specifies 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.

Default: 0

Limits: 1 to 3 numeric characters from 0 to 255

Batch keyword: TP1M

C

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 permitted.

TYPE2

Indicates that the following two fields (M and C) contain information about TYPE2 tape drives need for the job. No input is allowed for this field.

M

(Optional) Specifies 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.

Default: 0

Limits: 1 to 3 numeric characters from 0 to 255

Batch keyword: TP2M

C

Indicates the calculated value for the number of TYPE2 tape drives needed for the job. The 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 permitted.

More information:

DB.3 - Job Predecessor/Successor Menu Panel (see page 150)
#SCC Statement (see page 322)

Usage Notes

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

If different default values are wanted, a job with the wanted 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 wanted 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 want to control which jobs are eligible to be marked as LATE, you can use the LATEPROMPT= keyword on the OPTIONS statement in the CA Workload Automation SE initialization file. If you specify LATEPROMPT=LATE, jobs that are defined with PROMPT: N never show a status of LATE on LQ displays (including LREQ, LRDY, LACT, and so on). This lets your operations staff concentrate on important jobs when they show up as LATE.

Example: Batch Input

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
```

For more information about the batch keywords, see the previous definitions of these panel fields.

How You Change a CPU Job

The basic way to change the characteristics of a job, which appear on the DB.1 panel, is to use the UPD function. However, you can make wholesale changes to many jobs using the Batch Terminal Interface (BTI) facility.

Note: For more information about BTI, see the *Interface Reference Guide*.

The database information must always agree with the JCL for the job. Therefore, whenever permanent changes are made to the JCL, resynchronize the database with new JCL. The CA Workload Automation SE LOAD process accomplishes this process, 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 panel to Y if the change is made outside of CA Workload Automation SE (for example, through TSO).
- Make any permanent changes to the JCL through the DB.7 panel. This method causes the RELOAD field on the DB.1 panel to be set to Y automatically.

Note: The DB.3 panels establish dependency relationships for CPU jobs, workstation networks, and so forth, thereby defining prerequisites. Whenever the JCL changes for any job, consider whether those changes affect the dependency relationships for the job.

More information:

Batch Input (see page 35)
DB.3 - Job Predecessor/Successor Menu Panel (see page 150)
Mass Changes at Existing Site (see page 440)

DB.10 - XP Job Definition Panel

The DB.10 - XP Job Definition panel lets you enter or review data related to XPJOB jobs.

To display the panel, enter:

- A.A as the FUNCTION on the DB Menu panel.
- DB.10 as the FUNCTION on any other menu or formatted input panel.
- DB.10, XPJOB, or DB.A.A as a top line command.

To exit the panel:

- Press PF3 to return to the DB Menu panel except with the TSO/ISPF interface when PF3 has been assigned to another function.
- Move the cursor to the top line of the panel and enter a top line command if you want another function.

```
------------------------  CA-7 XP Job Definition  -------------------------
Function: xxxxxxxx  (Add, DD, Delete, DELPRRN, Format, List, Purge, Update)
Job: xxxxxxxx

System: xxxxxxxx  JOBNET: xxxxxxxx  Owner: xxxxxxxx UID: nnn
XP Node: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
XP EXEC: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
         xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
         xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
         xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
XP PARM: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
         xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Optional PARMLIB: xxxxxxxxxxxxxxxx  Member: xxxxxxxx Use-Ovrd-Lib:

EXEC: x     SUTYPE: x     Cond-Code: xxxx RO: xx     DRClass: xxxxxxxx
Hold: x     Verify: x     Satisfaction Lead Time: nn  ARFSET: xxxxxxxx
Clock Time: nnnn     WLBClass: x     WLBPRTY: nnn      Trace: x
Don't Schedule Before: yyddd hhmm  After: yyddd hhmm  Retain: x
LTERM: xxxxxxxx  Prompt: x  Rqmt List: x Rqmts Not Used: x

PROGRAM: SM22   MSG-INDX: nn   -- DB.10    --   yy.ddd / hh:mm:ss
MESSAGE: ENTER FUNCTION, TRANSFER OR ENTER A COMMAND ON THE TOP LINE
```

This panel contains the following fields:

FUNCTION

Specifies the function to perform. Value must be the name of some other panel or one of the following:

Add

Adds a job to the database if the same-named job does not currently exist.

DD

Deletes a job and its associated data from the database (documentation, schedules, and so forth).

Delete

Deletes a job and its associated data from the database (documentation, schedules, and so forth).

DELPRRN

Deletes the parameter and associated data from the prior run queue for this job.

Format

Clears the panel of all input data.

List

Lists a job and job-related information. In batch, a formatted panel is not listed; only a found or not found message is returned.

Purge

Same as delete, but also deletes job trigger definitions that trigger the job being deleted and job requirement definitions that require the job being deleted. This includes all virtual resources connected to the job.

Update

Updates database information about a job.

Batch keyword: Positional parameter

Note: A default interpretation can be set for the DELETE function in the CA Workload Automation SE initialization file. For more information, see the OPTIONS statement in the chapter "Initialization" of the *Systems Programming Guide*.

Job

Defines the job name on which to perform the indicated function.

Limits: 1 to 8 alphanumeric characters. The job name format of UCC7R*xxx* (where *xxx* is any three 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 five characters.

Batch keyword: Positional parameter

System

(Optional) Defines the user-defined application system name of which this job is a part. This field will be transmitted to the cross-system platform as the JOBSET name. If this field is not completed, the JOBSET field is the value of the CA Workload Automation SE XPS monitor name.

Limits: 1 to 8 alphanumeric characters. This field cannot contain a comma.

Batch keyword: SYSTEM

JOBNET

(Optional) Defines the name of a job network of which this job is a part.

Note: This field is not transmitted to the cross-system platform but may be used within CA Workload Automation SE internally.

Limits: 1 to 8 alphanumeric characters. This field cannot contain a comma.

Batch keyword: JOBNET

Owner

(Optional) Defines the ID identifying ownership of this job. This field can have multiple meanings, depending on the CA Workload Automation SE startup options specified on the XPDEF statement. This field can be used to identify the "User ID" resource record (entered through the XPSWD,OWNER= command) in which to find the user ID and password for the job being executed on the alternate platform. If the XPDEF definition permits this field to code the user ID, the user ID under which the job executes is obtained from here, and a password is not associated with the job on the alternate platform. If this field is left blank and the XPDEF definition allows USER, the security information must be specified in the parameter library. If PSWDLOC specifies NODE, the user ID and password are obtained from the Node Table information. If no user ID is found for the AJB, the job is submitted to the platform without any user information (the platform may reject the job execution if security is set that way).

Limits: 1 to 8 alphanumeric characters. 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.

Batch keyword: OWNER

UID

(Optional) Defines the CA Workload Automation SE user security identification.

Default: 0 (No internal security identification)

Limits: 1 to 3 numeric characters from 0 to 255

Batch keyword: USERID

XP Node

Defines the CAICCI node to which the execution of this job is targeted. This should state the primary node. If this node is unavailable and the node definition has alternate nodes defined, the execution of this job can be directed to an alternate node.

Limits: 1 to 44 alphanumeric characters, although current z/OS CAICCI restricts this name to eight characters.

Batch keyword: NODE

XP EXEC

Defines the executable (file, command) to execute at the targeted cross-platform node. If the targeted platform is a mainframe scheduling system such as CA Workload Automation SE, this indicates the job name to be executed on that platform. If the targeted platform is CA Workload Automation AE system, this indicates a job defined in the CA Workload Automation AE system. If this field begins and ends in single quote (') marks, these are removed before being passed in the transmission data. Otherwise, the field is passed to the target system *as-is* and in its entirety. This includes any embedded blanks and quotation marks.

Limits: Three 64-byte lines and one 52-byte line, comprising 244 alphanumeric characters, and file delimiters of forward slash (/) and backward slash (\) signs.

Batch keyword: EX1, EX2, EX3, EX4

Note: Some EBCDIC characters do not translate to ASCII, and thus the file name must be carefully defined so that translation can occur correctly between EBCDIC and ASCII.

You must enclose the value of the batch parameter in parentheses if this field contains commas or trailing spaces and you are using the batch method to add or update the XP EXEC or XP PARM fields.

XP PARM

(Optional) Defines up to 128 bytes of parameter data to pass to the file or command being executed on the alternate platform. This data is supplied as PARM1 to the data being transmitted to the alternate platform. If this field begins and ends in single quote (') marks, these are removed before being passed in the transmission data. This value may be overridden if the PARMLIB/MEMBER field PARM1 is coded.

Limits: Two 64-byte lines, comprising up to 128 EBCDIC characters.

Batch keyword: XP1 and XP2

Note: Some EBCDIC characters do not translate to ASCII, and thus the file name must be carefully defined so that translation can occur correctly between EBCDIC and ASCII.

Optional PARMLIB

(Optional) Defines an additional, optional indexed or symbolic (PDS) library from where execution data to be associated with this job may be found. This information is normally PARM1 through PARM64 keywords but can vary based on CA Workload Automation SE system configuration options. If PARM1 field is found within this file, it overrides the data specified in the XP PARM field listed on the panel.

Limits: 1 to 16 alphanumeric characters. If this field starts with a numeric value, the value is treated as a JCL Index value (0 - 253), and the associated numbered JCL library, as defined in the initialization file JCL statement, is used. If the first character is not numeric, the field must start with an ampersand (&) and denote the library variable as defined using a /JCL command.

Batch keyword: PRMLIB

Note: Specifying data in the PARMLIB/MEMBER fields is in addition to anything specified in the XPPARM field. If XPPARM is coded, and if the PARMLIB/Member contains a PARM1 statement, the PARMLIB specification overrides the XP PARM field. Thus, if both XPPARM and PARMLIB PARM1 fields contain data, the data obtained from the PARMLIB/MEMBER is used in the data transmitted to the target node.

Member

(Optional) Indicates the PDS member name of the preceding library where the parameters for this job reside.

Default: Same as the Job field

Limits: 1 to 8 alphanumeric characters, beginning with an alphabetic character

Batch keyword: MEMBER

Use-Ovrd-Lib

(Optional) Specifies whether to retrieve the JCL 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

Batch keyword: JOVRDLIB

EXEC

(Optional) Indicates whether to execute this job.

Y

Specifies to execute the job. This is the default.

N

Specifies that the job does not run but shows a normal completion as if it did run.

Note: Consider the following items:

- Since nonexecutable jobs do not get executed, no data sets are created. This means that any jobs waiting for data set requirement that may have been created for the job execution are not created and thus jobs may wait on these data set requirements.
- VRM definitions are ignored for the nonexecutable jobs.
- Critical Path Management flows cannot be started with nonexecutable jobs and thus the critical path may not be monitored.
- ARF recovery is not invoked for nonexecutable jobs.

Batch keyword: EXEC

SUTYPE

(Optional) Specifies the type of "switch user" command to execute at the UNIX target node.

Y

Executes an "SU -" causing the environment set up to include execution of the ".PROFILE" for the target user. This is the default.

N

Executes an "SU" command without the profile option.

Batch keyword: SUTYPE

Cond-Code

(Optional) Used with RO (relational operator), defines the job-level condition codes used to determine whether a job executes successfully. All condition codes returned from an XPJOB job are treated as positive values. Any negative value returned is converted to an absolute (positive) value.

Default: 0

Limits: 1 to 4 numeric characters from 0 to 9999

Batch keyword: CONDCODE

RO

(Optional) Specifies the relational operator of the condition code (COND-CODE).

The following are the possible values:

0

Performs no condition test. This is the default.

EQ

Specifies equal to.

LT

Specifies less than.

GT

Specifies greater than.

GE

Specifies greater than or equal to.

LE

Specifies less than or equal to.

NE

Specifies not equal to.

IG

Performs no evaluation of the job. CA Workload Automation SE always assumes the job completes successfully, regardless of condition codes, abend codes, or runtime JCL errors.

Batch keyword: RELOPR

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 condition code.

Note: This test is for internal use by CA Workload Automation SE only. It simply tells CA Workload Automation SE what action to take *after* the job completes.

DRClass

(Optional) Defines the job's disaster recovery class. This field has no impact on processing during normal execution. When running in disaster recovery mode, the disaster recovery class is used to determine whether the job should execute.

Note: For more information about disaster recovery mode, see the *Systems Programming Guide*.

Limits: 1 to 8 alphanumeric characters. Values must start with a letter, #, or $ (not @) and can include letters, #, $, @, and numbers. Disaster recovery classes cannot contain embedded blanks.

Batch keyword: DRCLASS

Hold

(Optional) Specifies whether to put this job in a hold status when it enters the request queue.

N

Does not put this job in a hold status. This is the default.

Y

Puts this job in a hold status.

Batch keyword: HOLD

Verify

(Optional) Specifies whether this job requires any presubmission manual verification.

Note: This function is similar to VERIFY command.

N

Specifies that this job does not require presubmission manual verification. This is the default.

Y

Specifies that this job requires presubmission manual verification.

Batch keyword: VERIFY

Satisfaction Lead Time

(Optional) Defines the number of hours to consider when satisfying job dependent requirements.

The following are the possible values:

0

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

99

Indicates that the requirement is never 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

Indicates that each predecessor job must have run within the last *nn* hours since the last run of this job.

Limits: 1 - 98

Batch keyword: JOBLDTM

Note: For XPJOB jobs, JOB and DSN Satisfaction Lead Time are treated as one.

ARFSET

(Optional) Defines the collection of ARF definitions that apply to this job. Remember that ARF recovery is not invoked for nonexecutable jobs.

Limits: 1 to 8 alphanumeric characters. The specified ARFSET must be defined in the ARF database.

Batch keyword: ARFSET

Clock Time

(Optional) CA Workload Automation SE maintains certain feedback data in its database, including a weighted average elapsed execution time. If the database is updated with a clock time of 0000, then in addition to this clock time field the counters for the number of runs, number of times late, and number of restarts is also set to 0000. One use of this value is deadline prompting. If 2359 is specified, deadline prompt time is not adjusted. The deadline time remains due-out time minus lead time.

Note: Clock time average is not updated if the job is not successful in its execution. The weighted average is the value in the database times five, plus the value from the current execution, divided by six. This tends to keep the current run from possibly skewing the value.

Default: 0000

Limits: 4 numeric characters specified as *hhmm*, where *hh* can be 00 through 23, and *mm* can be 00 through 59

Batch keyword: ELAPTM

WLBClass

(Optional) Defines the CA Workload Automation SE WLB job class. If Workload Balancing is used, 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.

Default: A

Limits: 1 alphanumeric character

Batch keyword: CLASS

WLBPRTY

(Optional) Defines the CA Workload Automation SE 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.

Default: 100 if WLB is used or 0 is WLB is not used

Limits: 1 to 3 numeric characters from 0 to 255

Batch keyword: PRTY

Trace

(Optional) Indicates whether to trace the activity associated with this job as it moves through the CA Workload Automation SE queues. The possible values include:

N

Do not activate tracing for this job. This is the default.

Y

Issue WTOs as the job enters into the submission process and retain up to 256 characters of the data sent for inclusion in a log record.

Batch keyword: XPTRACE

Don't Schedule Before

(Optional) Specifies that this job is not to be scheduled before this date and time. This keyword is especially helpful for timing the start of new jobs.

Panel input is:

BEFORE: *yyddd hhmm*

Batch input is:

BDATE=*yyddd*,BTIME=*hhmm*

Default: All zeros

Limits: Julian date specified as *yyddd* and time specified as *hhmm*

Batch keyword: BDATE and BTIME

After

(Optional) Specifies that this job is not to be scheduled after this date and time. This keyword is especially helpful for timing the permanent discontinuation of a job.

Panel input is:

AFTER: *yyddd hhmm*

Batch input is:

ADATE=*yyddd*,ATIME=*hhmm*

Default: 99999 0000

Limits: Julian date specified as *yyddd* and time specified as *hhmm*

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=09031 and BTIME=2359 and ADATE=09001 and ATIME=0000, the job does not schedule during the time from Jan. 1, 2009 through Jan. 31, 2009, but does schedule at all other times. CA Workload Automation SE uses current date and time for comparisons.

Retain

(Optional) Specifies whether to retain the job data sent to the target node in the trailer queue after a successful execution.

N

Do not send the job data. This is the default.

Y

Send the job data.

Default: N

Batch keyword: RETJCL

LTERM

(Optional) Defines a logical terminal name to receive messages about this job.

Default: MASTER

Limits: 1 to 8 alphanumeric characters

Batch keyword: LTERM

Note: Although a CPU job can specify LTERM either on the JOB (DB.1) Definition panel or in the initialization file JCL statement from which the JCL is pulled, for XPJOBs, the only place to define the terminal to which messages are sent is on the XPJOB (DB.10) panel.

Prompt

(Optional) Specifies whether to issue prompt messages if this job is late.

Y

Issue the prompt messages. This is the default.

N

Do not issue the prompt messages.

Batch keyword: PROMPT

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

Rqmt list

(Optional) Specifies whether to list the pre-execution requirements for this job when it enters the request queue.

Y

List the requirements. This is the default.

N

Do not list the requirements.

Batch keyword: RQLST

Rqmts Not Used

(Optional) Specifies whether to issue error messages for job requirements not used.

Y

Issue error messages. This is the default.

N

Do not issue error messages.

Batch keyword: RQMSG

Usage Notes

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

If different defaults are desired for an XPJOB job, set the initialization file DBASE statement with the DEFAULTXPJ= parameter. The job name specified in this parameter will set the defaults for all subsequently added XPJOB jobs.

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

Example: Batch Input

This batch input example adds XPJOB job CA07XP01, specifying the system name, node, executable (EX1) and PARM01 (XP1):

```
XPJOB
ADD,CA07XP01,SYSTEM=TESTXPJB,
NODE=TESTPLN,
EX1=(C:\NSM\BIN\CAU9TEST),
XP1=(T=60,F=12)
```

DB.11 - Agent Job Definition Panel

The DB.11 - Agent Job Definition panel lets you enter or review data related to agent jobs.

To display the panel, enter:

- A as the FUNCTION on the DB Menu panel; this leads to another menu for job type selection.
- DB.11 as the FUNCTION on any other menu or formatted input panel.
- DB.11 or AGJOB as a top line command.

To exit the panel:

- Press PF3. If you entered the agent job command (excluding menu commands), you are returned to the DBM Menu panel except with the TSO-ISPF interface when the PF3 key is assigned to another function. If you entered the DB panel, you are returned to the menu from which you can directly invoke the agent.
- Move the cursor to the top line of the panel and enter a top line command for another function.

When you enter a DB panel identifier for a valid agent job type, or traverse through the menus, the CA-7 Agent Job Definition panel is displayed with the Agent Job Type field pre-filled. If you enter DB.11 or AGJOB, the Agent Job Type field is *not* pre-filled.

If you use the Format command to clear the screen and want to add a new agent job definition, you must supply the Agent Job Type.

Note: You *can* change the agent job type. Be careful that you do not change it to a value that is *not* valid for the type of job you want to execute.

```
------------------------  CA-7 Agent Job Definition  -----------------------
Function:           (Add, DD, Delete, DELPRRN, Format, List, Purge, Update)
Job:

System:           JOBNET:           Owner:          UID:

Agent Job Type: NT_JOB
Agent:
User:

Parmlib:                            Member:          Use-Ovrd-Lib:

EXEC:             Hold:             Verify:
DRClass:          ARFSET:           Satisfaction Lead Time:
WLBClass:         WLBPRTY:          Clock Time:
Don't Schedule Before:              After:
LTERM:            Prompt:           Rqmt List:     Rqmts Not Used:



PROGRAM: SM23   MSG-INDX: nn   -- DB.11    --   yy.ddd / hh:mm:ss
MESSAGE: ENTER FUNCTION, TRANSFER OR ENTER A COMMAND ON THE TOP LINE
```

This panel contains the following fields:

Function

Specifies the function to perform. Value must be the name of some other panel or one of the following:

Add

Adds a job to the database if the same-named job does not currently exist.

DD

Deletes a job and its associated data from the database (documentation, schedules, and so forth).

Delete

Deletes a job and its associated data from the database (documentation, schedules, and so forth).

DELPRRN

Deletes the parameter and associated data from the prior run queue for this job.

Format

Clears the panel of all input data.

List

Lists a job and job-related information. In batch, a formatted panel is not listed; only a found or not found message is returned.

Purge

Same as delete, but also deletes job trigger definitions that trigger the job being deleted and job requirement definitions that require the job being deleted. This includes all virtual resources connected to the job.

Update

Updates database information about a job.

Batch keyword: Positional parameter

Note: A default interpretation can be set for the DELETE function in the CA Workload Automation SE initialization file. For more information, see the OPTIONS statement in the chapter "Initialization" of the *Systems Programming Guide*.

Job

Defines the job name on which to perform the indicated function.

Limits: 1 to 8 alphanumeric characters. The job name format of UCC7R*xxx* (where *xxx* is any three 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 five characters.

Batch keyword: Positional parameter

System

(Optional) Defines the user-defined application system name of which this job is a part. The internal agent job ID uses this field. If the System Name is blank, the word "SYSTEM" is used as part of the internal agent job ID.

Limits: 1 to 8 alphanumeric characters. This field cannot contain a comma.

Batch keyword: SYSTEM

JOBNET

(Optional) Defines the name of a job network of which this job is a part.

Note: This field is not transmitted to the cross-system platform but may be used within CA Workload Automation SE internally.

Limits: 1 to 8 alphanumeric characters. This field cannot contain a comma.

Batch keyword: JOBNET

Owner

(Optional) Defines the ID identifying ownership of this job. Depending on the CA Workload Automation SE startup options taken and the external security product being used, CA Workload Automation SE can offer this value to the external security package at job submission time as the authority for this job to execute.

Limits: 1 to 8 alphanumeric characters. 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.

Batch keyword: OWNER

UID

(Optional) Defines the CA Workload Automation SE user security identification.

Default: 0 (No internal security identification)

Limits: 1 to 3 numeric characters from 0 to 255

Batch keyword: USERID

Agent Job Type

Defines the type of agent job. The valid values are listed in the Agent Command/Job Type table found in the *Interface Reference Guide*.

The field has no default. If you use the menus (starting with DB.A or XPSMENU), the Agent Job Type is pre-filled based on your menu selection.

Batch keyword: AGJOBTYP

Agent

Identifies the explicit distributed platform where the job is to run.

Limits: 1 to 16 alphanumeric characters. The value must be defined to CA Workload Automation SE using the IASAGENT DD statement.

Batch keyword: AGENT

User

(Optional) Defines the user ID that is passed to the distributed platform where the job is to run.

Limits: Two 64-byte lines comprising 128 alphanumeric characters.

Batch keyword: AG1, AG2

Parmlib

Defines an indexed or symbolic (PDS) library from where execution data that is associated with this job can be found. This information is normally keyword parameters and values required by the particular agent job type.

Limits: 1 to 16 alphanumeric characters. If this field starts with a numeric value, the value is treated as a JCL Index value (0 - 253), and the associated numbered JCL library, as defined in the initialization file JCL statement, is used. If the first character is not numeric, the field must start with an ampersand (&) and denote the library variable as defined using a /JCL command.

Batch keyword: PRMLIB

Member

(Optional) Indicates the PDS member name of the preceding library where the parameters for this job reside.

Default: Same as the Job field

Limits: 1 to 8 alphanumeric characters, beginning with an alphabetic character

Batch keyword: MEMBER

Use-Ovrd-Lib

(Optional) Specifies whether to retrieve the JCL 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

Batch keyword: JOVRDLIB

EXEC

(Optional) Indicates whether to execute this job.

Y

Specifies to execute the job. This is the default.

N

Specifies that the job does not run but shows a normal completion as if it did run.

Note: Consider the following items:

- Since nonexecutable jobs do not get executed, no data sets are created. This means that any jobs waiting for data set requirement that may have been created for the job execution are not created and thus jobs may wait on these data set requirements.
- VRM definitions are ignored for the nonexecutable jobs.
- Critical Path Management flows cannot be started with nonexecutable jobs and thus the critical path may not be monitored.
- ARF recovery is not invoked for nonexecutable jobs.

Batch keyword: EXEC

Hold

(Optional) Specifies whether to put this job in a hold status when it enters the request queue.

N

Does not put this job in a hold status. This is the default.

Y

Puts this job in a hold status.

Batch keyword: HOLD

Verify

(Optional) Specifies whether this job requires any presubmission manual verification.

Note: This function is similar to VERIFY command.

N

Specifies that this job does not require presubmission manual verification. This is the default.

Y

Specifies that this job requires presubmission manual verification.

Batch keyword: VERIFY

DRClass

(Optional) Defines the job's disaster recovery class. This field has no impact on processing during normal execution. When running in disaster recovery mode, the disaster recovery class is used to determine whether the job should execute.

Note: For more information about disaster recovery mode, see the *Systems Programming Guide*.

Limits: 1 to 8 alphanumeric characters. Values must start with a letter, #, or $ (not @) and can include letters, #, $, @, and numbers. Disaster recovery classes cannot contain embedded blanks.

Batch keyword: DRCLASS

ARFSET

(Optional) Defines the collection of ARF definitions that apply to this job. Remember that ARF recovery is not invoked for nonexecutable jobs.

Limits: 1 to 8 alphanumeric characters. The specified ARFSET must be defined in the ARF database.

Batch keyword: ARFSET

Satisfaction Lead Time

(Optional) Defines the number of hours to consider when satisfying job dependent requirements.

The following are the possible values:

0

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

99

Indicates that the requirement is never 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

Indicates that each predecessor job must have run within the last *nn* hours since the last run of this job.

Limits: 1 to 98

Batch keyword: JOBLDTM

Note: For agent jobs, JOB and DSN Satisfaction Lead Time are treated as one.

WLBClass

(Optional) Defines the CA Workload Automation SE WLB job class. If Workload Balancing is used, 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.

Default: A

Limits: 1 alphanumeric character

Batch keyword: CLASS

WLBPRTY

(Optional) Defines the CA Workload Automation SE 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.

Default: 100 if WLB is used or 0 is WLB is not used

Limits: 1 to 3 numeric characters from 0 to 255

Batch keyword: PRTY

Clock Time

(Optional) CA Workload Automation SE maintains certain feedback data in its database, including a weighted average elapsed execution time. If the database is updated with a clock time of 0000, then in addition to this clock time field the counters for the number of runs, number of times late, and number of restarts is also set to 0000. One use of this value is deadline prompting. If 2359 is specified, deadline prompt time is not adjusted. The deadline time remains due-out time minus lead time.

Note: Clock time average is not updated if the job is not successful in its execution. The weighted average is the value in the database times five, plus the value from the current execution, divided by six. This tends to keep the current run from possibly skewing the value.

Default: 0000

Limits: 4 numeric characters specified as *hhmm*, where *hh* can be 00 through 23, and *mm* can be 00 through 59

Batch keyword: ELAPTM

Don't Schedule Before

(Optional) Specifies that this job is not to be scheduled before this date and time. This keyword is especially helpful for timing the start of new jobs.

Panel input is:

BEFORE: *yyddd hhmm*

Batch input is:

BDATE=*yyddd*,BTIME=*hhmm*

Default: All zeros

Limits: Julian date specified as *yyddd* and time specified as *hhmm*

Batch keyword: BDATE and BTIME

After

(Optional) Specifies that this job is not to be scheduled after this date and time. This keyword is especially helpful for timing the permanent discontinuation of a job.

Panel input is:

AFTER: *yyddd hhmm*

Batch input is:

ADATE=*yyddd*,ATIME=*hhmm*

Default: 99999 0000

Limits: Julian date specified as *yyddd* and time specified as *hhmm*

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=09031 and BTIME=2359 and ADATE=09001 and ATIME=0000, the job does not schedule during the time from Jan. 1, 2009 through Jan. 31, 2009, but does schedule at all other times. CA Workload Automation SE uses current date and time for comparisons.

LTERM

(Optional) Defines a logical terminal name to receive messages about this job.

Default: MASTER

Limits: 1 to 8 alphanumeric characters

Batch keyword: LTERM

Note: Although a CPU job can specify LTERM either on the JOB Definition (DB.1) panel or in the initialization file JCL statement from which the JCL is pulled, for agent jobs, the only place to define the terminal to which messages are sent is on the AGJOB (DB.11) panel.

Prompt

(Optional) Specifies whether to issue prompt messages if this job is late.

Y

Issues the prompt messages. This is the default.

N

Does not issue the prompt messages.

Batch keyword: PROMPT

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

Rqmt List

(Optional) Specifies whether to list the pre-execution requirements for this job when it enters the request queue.

Y

Lists the requirements. This is the default.

N

Does not list the requirements.

Batch keyword: RQLST

Rqmts Not Used

(Optional) Specifies whether to issue error messages for job requirements not used.

Y

Issues error messages. This is the default.

N

Does not issue error messages.

Batch keyword: RQMSG

Usage Notes

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

If you want different defaults for an agent job, set the initialization file DBASE statement with the DEFAULTAG parameter. The job name specified in this parameter sets the defaults for all subsequently added agent jobs.

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

Example: Batch Input

This batch input example adds UNIX_JOB CA07UNX1 specifying the system name, agent job type, agent name, user ID, and PARMLIB.

```
AGJOB
ADD,CA07UNX1,SYSTEM=TESTAGJB,
AGJOBTYP=UNIX_JOB,
AGENT=UNIX_NY,
AG1=userid1,
PRMLIB=200
```

Chapter 3: Scheduling Work

This section identifies the panels and commands used to schedule work in CA Workload Automation SE.

This section contains the following topics:

Work Scheduling (see page 83)
Schedule Definition (see page 91)

Work Scheduling

CA Workload Automation SE uses both date/time-driven and event-driven facilities to cause work to be scheduled and initiated. To CA Workload Automation SE, work is defined as jobs, preprocessing and postprocessing workstation networks, or both. Event-driven scheduling, known as trigger scheduling, is a far more efficient technique than date/time oriented schedules.

For 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 Workload Automation SE system. For postprocessing networks, output workstations are scheduled based on the scheduling of the 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 can be used in combination when scheduling and initiating work. For example, a job can have a timed schedule for processing but can 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 on, 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 can be chosen for a single schedule ID (for example, daily and monthly). However, a job that is scheduled automatically on a defined date/time basis is not scheduled more than once per day for a given schedule ID.

More information:

Event Scheduling with Triggers (see page 88)

Date and Time Scheduling

The schedule scan function performs date/time-driven scheduling based on initialization file parameters and database definitions. By combining base calendar and schedule definitions through a resolution process, CA Workload Automation SE stores information about date/time related work in its database.

Base Calendars

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

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

Either batch or online facilities can generate base calendars. To use the online facility (DB.2.8), a CA Workload Automation SE calendar PDS must be allocated and identified to CA Workload Automation SE. Processing of base calendars for resolution of schedules is the same regardless of whether they were generated using the batch or online facilities.

Note: For information about enabling online calendar maintenance, see the *Systems Programming Guide*.

The CA Workload Automation SE CALENDAR macro can generate calendars. 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 Workload Automation SE CAL2LOAD or a specified calendar library, and are stored in the form of bit masks that 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 calendars should be generated by July 1 of the current year. New base calendars can 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 that reflects the beginning and ending days of each month, holidays, or any other nonprocessing days that were defined when the calendar was produced.

Note: For more information about the PRINT command, see the *Command Reference Guide*.

Perpetual Calendars

Using perpetual calendars can automate the generation of base calendars. When the criteria for building a calendar are defined in the Perpetual Calendar DSN (PCALDSN), the corresponding base calendar (SCAL*yyxx*) is generated and used on the first reference of the calendar if the calendar does not exist. The reference to the calendar could be a PRINT, RESOLV, CALMOD, or any other command or process that needs a calendar.

Perpetual calendars, defined with the member name PCALYY*xx*, describe scheduling and nonscheduling days in English-like terms. These members can be used each year, because the command that references a calendar builds the SCAL*yyxx* member if it does not already exist. This permits the same set of criteria to be used each time. As a result, perpetual calendars are a tool for jobs that use the same "relative" processing days year after year. For example, a job that executes every Monday, Wednesday, and Friday can reference a perpetual calendar PCALYY3W that has the following statements:

```
Set all days nonscheduled

Set Mondays Wednesdays Fridays schedule
```

When a command such as PRINT is used, CA Workload Automation SE generates the base calendar SCAL*yy*3W (as an example) that sets the processing days for each Monday, Wednesday, and Friday of each week. This eliminates the need to set up each year the corresponding SCAL*yy*3W.

Note: For more information about perpetual calendars see the *Systems Programming Guide*.

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 panels. The result of this resolution is the schedule used by CA Workload Automation SE to schedule preprocessing work and jobs. The following figure 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.

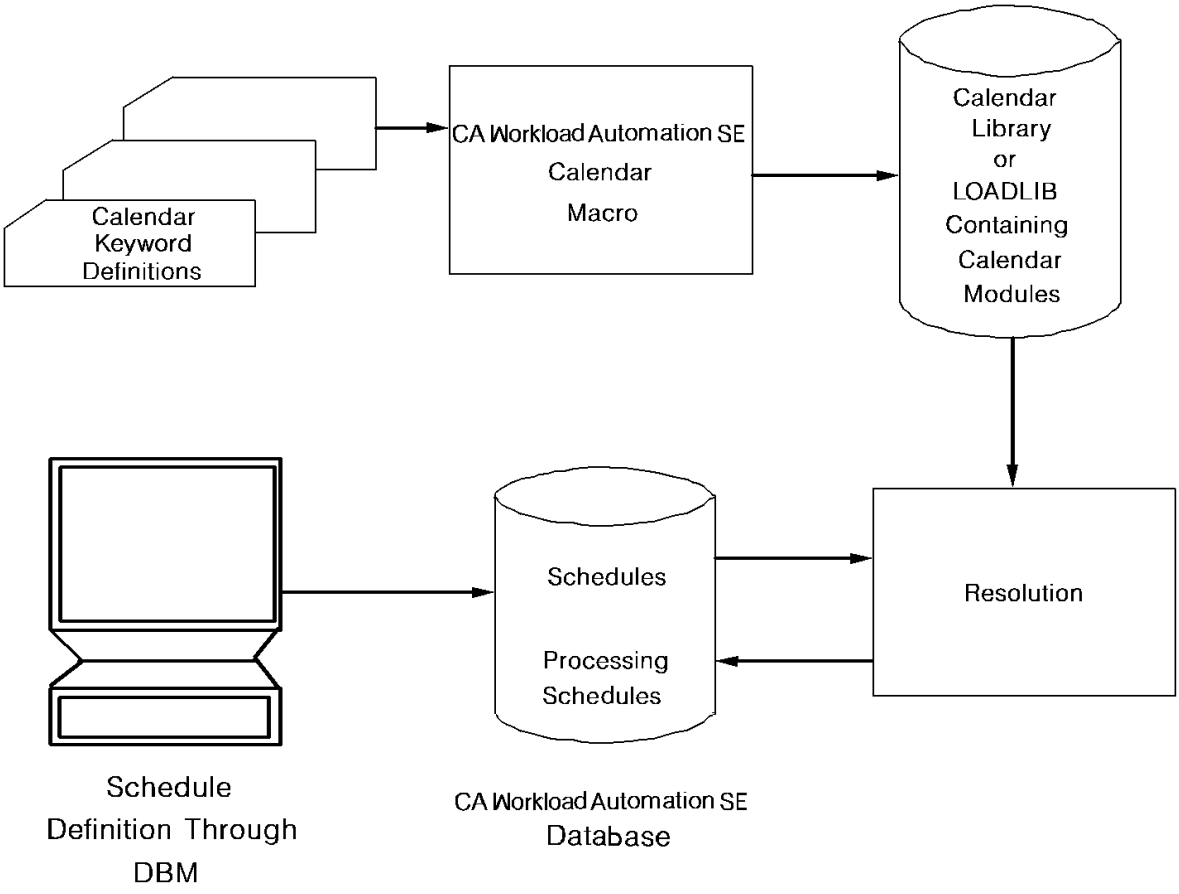
RESOLV Command

Use the top line RESOLV command to create or modify processing schedules for jobs or workstation networks that are to be scheduled on a date/time basis. Work that is scheduled by a trigger or on-request work that 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. The RESOLV command provides options for resolving all or any subset of those tasks scheduled with a single command.

Note: For more information about the RESOLV command, see the *Command Reference Guide*.

The following figure illustrates calendar and schedule creation and resolution.



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, time, or both 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 panel. This panel displays a SCHID in calendar format showing when the job or network does or does not run. The schedule can be temporarily modified until the next DB.2.7 panel or RESOLV command is used against it.

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

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 that 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.

Event Scheduling with 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 can 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. Three types of triggers are available in CA Workload Automation SE:

- Job
- Network
- Data Set

With each trigger, you can trigger multiple jobs if you want.

Job Triggers

When a predecessor job completes successfully, a trigger initiates scheduling of the successor or dependent jobs that will be brought into the request queue. This newly triggered job can 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 jobs that 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 jobs to be triggered are 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, 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 panels.

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

Data set triggers are very useful if you have a job that 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 panel 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, 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 recreated, the trigger occurs again.

On-Request Scheduling

Provisions are included in CA Workload Automation SE to permit on-request scheduling of work. This is the method of scheduling jobs that 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.

DEMAND Command

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

- A DEMAND (or DEMANDH) command
- Job name
- JCL library ID (only needed if the CPU job is not already defined in the database; not valid with cross-platform jobs)

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

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.

RUN Command

The RUN command is used to request a job without verifying the availability of input requirements or performing the updates that 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 Workload Automation SE class 9.

The following normal scheduling activities of CA Workload Automation SE 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.

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.

Schedule Definition

Definition of work schedules is accomplished through DB.2 panels. 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 calendars is used to complete the definition of the processing schedule.

DB.2 - Scheduling Menu Panel

The DB.2 - Scheduling Menu panel lets you access the scheduling definition formatted panels.

To display the panel, enter:

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

To exit the panel:

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

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

```
--------------------------  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:
```

DB.2.1 - Job Scheduling Panel

Note: Although this panel and command specifically mentions CPU jobs, these functions are also available with all cross-platform jobs (for example, XPJOBs or agent jobs).

The DB.2.1 - Job Scheduling panel lets you define or review options taken for CPU jobs with date/time schedules.

To display the panel, enter:

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

To exit the panel:

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

```
------------------------  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:
```

This panel contains the following fields:

FUNCTION

Specifies the function to perform. Value must be the name of some other panel or one of the following:

CLEAR

Clears the panel 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 Job Scheduling Parameter Edit panel, which enables 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 panel. 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 panel.

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 can reference one job name while the SAVE or REPL function specifies a different job name. This lets the user make some changes to the schedule of an existing job and save those changes as the schedule for a new job.

Batch keyword: Positional parameter

n/a

Contains a constant value of JOB for jobs.

Limits: Required for batch only.

Batch keyword: Positional parameter

JOB

Specifies the required job name for which schedule information is being defined.

Limits: 1 to 8 alphanumeric characters

Batch keyword: JOB

SCAL

Specifies the last two characters of the default base calendar ID to use for schedule resolution. The complete calendar ID used is in the format SCAL*nnxx* 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 panel.

Limits: 2 alphanumeric characters

Required for the SAVE, REPL, and RESOLV functions.

Optional for CLEAR, DELETE, and EDIT functions.

Automatically supplied for FETCH or FE functions.

Batch keyword: SCAL

SCHID-COUNT

Identifies a system-generated field that shows how many schedule IDs currently exist in the Edit Work File.

Usage Notes

You must resolve schedules defined on this panel against the specified base calendar with the RESOLV function or top line command before CA Workload Automation SE automatic scheduling can commence.

Job Scheduling Parameter Edit Panel

Note: Although this panel and command specifically mentions CPU jobs, these functions are also available with all cross-platform jobs (for example, XPJOBs or agent jobs).

The Job Scheduling Parameter Edit panel lets you define or review schedules for jobs that are run on a date and time basis.

To display the panel, enter EDIT as the function on the DB.2.1 panel.

To exit the panel:

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

```
------------------  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    --------------------REPEAT---------------------
           hhmm hhmm hhmm    INTERVAL: hhmm TYPE: xxxxx COUNT: nnnn STOP: 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: ____________________________________________________
                                                        DEFAULT SCAL: xx
-- _ -- SYMETRIC  START: ___   SPAN: ___                 SCHID-COUNT: nnn

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

This panel contains the following fields:

FUNCTION

Specifies the function to perform. Value must be the name of some other panel 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 panel 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 panel of user input data.

LIST

Lists all existing SCHIDs, one per panel.

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 panel. Work area data (changes) are retained but the database is not updated.

SR

Combination function that results in a SAVE, a return to the DB.2.1 panel, 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 Workload Automation SE database. If not specified, the scheduling changes are lost once leaving this panel. Entering the SR function automatically returns you to the prior scheduling panel.

SS

Combination function that results in a SAVE, a return to the DB.2.1 panel, and a subsequent SAVE to add a newly defined schedule in the database.

Batch keyword: Positional parameter

JOB

Reflects the job name from the previous panel. This is informational only.

SCHID

Specifies the numeric schedule ID on which the user wants to perform the specified action.

Limits: 1 to 3 numeric characters from 1 through 255

Batch keyword: SCHID

SCAL

(Optional) Specifies the override of the default SCAL for this SCHID. If this SCHID is using the default, this field is blank.

Limits: 2 alphanumeric characters

Batch keyword: SCAL

ROLL

(Optional) Specifies the action to take 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:

B

Roll the schedule back to the previous available processing day in the Base Calendar.

F

Roll the schedule forward to the next available processing day.

N

Do not roll. Schedule day must stand.

D

Do not roll and do not schedule.

Default: D

Batch keyword: ROLL

INDEX

(Optional) 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.

Limits: 4 numeric characters specified as I*nnn* where I can be plus (unsigned) or minus (-) and *nnn* can be 0 to 365 days

Batch keyword: INDEX

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.

DOTM

Specifies the due-out time of day for this schedule ID.

Limits: 4 numeric characters specified as *hhmm* where *hh* can be 00 through 24 and *mm* can be 00 through 59. Must be greater than 0000.

Required for ADD and REPL functions.

Optional for DELETE, EXIT, FORMAT, LIST, SAVE, SR, and SS functions.

Batch keyword: TIME

In batch format, this field and the next two fields must be provided as a list of *hhmm* values as follows:

TIME=(*dueout,lead,submit*)

LDTM

Specifies 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 panel is set to 2359.

Limits: 4 numeric characters specified as *hhmm* where *hh* can be 00 through 24 and *mm* can be 00 through 59. Must be greater than 0000.

Required for ADD and REPL functions.

Optional for DELETE, EXIT, FORMAT, LIST, SAVE, SR, and SS functions.

Batch keyword: TIME

SBTM

(Optional) Specifies 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 is 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.

Note: If the submit time is after the due-out time, the submit day value is set to the previous day.

Limits: 4 numeric characters specified as *hhmm* where *hh* can be 00 through 24 and *mm* can be 00 through 59

Batch keyword: TIME

INTERVAL

(Optional) Specifies that the job should be repeated (executed more than once) and specifies the amount of time between each iteration.

If INTERVAL is specified, the SBTM (submit time requirement) and TYPE fields are required.

If both INTERVAL and COUNT are specified, the INTERVAL value times the COUNT value must total less than 24 hours.

If INTERVAL is not specified, the job is not repeated.

Note: For more information about how deadline, due-out, and submit time requirements are calculated for repeating jobs, see the Usage Notes.

Limits: 4 numeric characters in *hhmm* format where *hh* can be from 0 to 23 and *mm* can be from 00 to 59

Batch keyword: INTERVAL

TYPE

(Optional) Determines how the submit time requirement is calculated for repeating jobs. A TYPE of CLOCK indicates that the INTERVAL is added to the previous iteration's submit time requirement to determine the new submit time requirement. A TYPE of START and END indicates that the INTERVAL is added to the previous iteration's last (most recent) start and end times, respectively, to determine the new submit time requirement. TYPE is required if INTERVAL is specified. TYPE is discarded if INTERVAL is not specified.

Limits: CLOCK, START, or END

Batch keyword: TYPE

COUNT

(Optional) Specifies the maximum number of times to repeat the job. COUNT is discarded if INTERVAL is not specified. If both COUNT and STOP are specified, the job stops repeating when either the COUNT reaches zero or the STOP time is reached, whichever comes first. If COUNT is not specified, the job repeats until the STOP time is reached.

Limits: 1 to 4 numeric characters from 0 to 1439. Leading zeros can be discarded.

Batch keyword: COUNT

STOP

(Optional) Specifies the clock time on or after which the job should not be repeated. STOP is discarded if INTERVAL is not specified. If both COUNT and STOP are specified, the job stops repeating when either the COUNT reaches zero or the STOP time is reached, whichever comes first.

If the STOP time is less than the submit time requirement, the stop date is assumed to be the following day.

A stop time of zeros is the same as not specifying a value.

Default: 1439 minutes (24 hours minus 1 minute) after the submit time

Limits: 4 numeric characters specified as *hhmm*
where *hh* can be from 0 to 23 and *mm* can be from 00 to 59

Batch keyword: STOP

DAILY

(Optional) Specifies that the user wants to define a daily schedule. Daily means every available processing day as defined by the Base Calendar. Enter an X or Y to select this. If DAILY is used, the ROLL function has no effect.

Batch keyword: DAILY

WEEKLY

(Optional) Specifies that the user wants to define a weekly schedule. Enter an X or Y to select this. If WEEKLY is used, the run days of the week must be selected using the SUN through SAT fields.

Batch keyword: WEEKLY

SUN thru SAT

Defines specific days of the week on which to run the job.

Online:

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

Limits: Required if WEEKLY is used.

Batch keyword: DAYS

Batch:

Place an X in any of the seven 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

(Optional) Specifies that the user wants to define a monthly schedule. If this field is used, the user can 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. Enter an X or Y to select this.

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

Batch keyword: MONTHLY

JAN thru DEC

(Optional) Online:

Place an X in panel 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

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 to run the job. 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)

Limits: 1 to 14 numeric characters and required if MONTHLY and DAY-OF-WEEK are used.

Batch keyword: WEEKS

Batch:

WEEKS=1 -0 (separated by blanks)

DAY-OF-WEEK

Specifies which days of the week to run the job. If used, each day must be the first three letters of the wanted run days just as it appears on the panel following the weekly field (for example, SUN, MON, and so on).

Online:

DAY-OF-WEEK:TUE,THU
(separated by blanks or commas)

Limits: 3 to 27 alphanumeric characters and required only if WEEK is used with MONTHLY.

Batch keyword: DOW

Batch:

DOW=TUE THU (separated by blanks)

RDAY

(Optional) 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 you have days of the month when this job is not to run, a slash (/) can be used with the unsigned or negative values. Valid positive values range from 1 to 31. Valid negative values range from -0 to -30.

Limits: 1 to 60 numeric characters

Batch keyword: RDAYS

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

(Optional) Defines an annual schedule. To select this type of schedule, enter an X or Y in this optional field.

Limits: If ANNUAL is used, DAY is required.

Batch keyword: ANNUAL

DAY

(Optional) Specifies on which days of the annual schedule the user wants 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.

Limits: 1 to 55 numeric characters and required 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

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

SYMETRIC

(Optional) Used with the START and SPAN fields, defines 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 that cannot exceed eight characters.) The selection of this type of schedule is made by entering an X or Y.

Batch keyword: SYMETRIC

START

(Optional) Used with SPAN and SYMETRIC fields, defines 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 is 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.

Limits: 3 numeric characters specified as Julian day of the year from 1 to 365

Batch keyword: START

SPAN

(Optional) Used with SYMETRIC and START, defines 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 *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.

Limits: 1 to 3 numeric characters from 1 to 255

Batch keyword: SPAN

SCHID-COUNT

Identifies a system-generated field containing the current schedule ID count for this job.

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 one hour.

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

Usage Notes for Repeating Jobs

A job can be brought into the request queue once through schedule scan and executed more than once. Each iteration must successfully complete (or be forced complete) before the next iteration is added to the request queue. The following fields are used with repeating jobs:

INTERVAL

Required. Specifies the amount of time between each iteration of the job.

TYPE

Required. Specifies how the submit time requirement of each iteration is calculated-by adding the INTERVAL to the previous iteration's submit time requirement (CLOCK), START, or END time.

COUNT

Optional. Specifies the maximum number of times the job will be repeated.

STOP

Optional. Specifies the clock time after which the job is not to be repeated. If not specified then the stop time will be 1439 minutes (24 hours minus one minute) after the submit time requirement of the first iteration.

DOTM

Required. Specifies the due-out time (expected completion time) of the first iteration.

LDTM

Required. Specifies the amount of time required to process the job.

SBTM

Required. Specifies the submit time requirement of the first iteration.

As each iteration successfully completes (or is forced complete), CA Workload Automation SE adds the INTERVAL to the submit time requirement, start time, or end time based on the TYPE setting. This value becomes the next iteration's submit time requirement.

The new submit time requirement is then compared to the STOP time. If either the new submit time requirement or the current time are greater than the STOP time then the job is not repeated.

The new job's due-out time is calculated as the new submit time requirement plus the difference between the original due-out time and the original submit time requirement.

The new job's deadline time is calculated as the new due-out time minus the original lead time (LEADTM).

The start and end times of the job are local to the system where the job executed, not local to the system where CA Workload Automation SE is executing. TYPE=START and END should be used with care for NJE and cross-platform jobs.

Once a repeating job is in the request queue, you can skip iterations by adjusting the job's submit time using the QM.3 (XUPD) command. For example, suppose a job is set to repeat every hour on the hour, but today only you want to skip the 11 a.m. and 12 noon iterations. After the 10 a.m. iteration completes, you can use QM.3 command to edit the job's early start time to 1300 (1 p.m.). The job then waits until 1 p.m. before continuing to repeat.

DB.2.2 - Input Network Scheduling Panel

The DB.2.2 - Input Network Scheduling panel lets you define or review scheduling options for input networks.

To display the panel, enter:

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

To exit the panel:

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

```
----------------------  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:
```

This panel contains the following fields:

FUNCTION

Specifies the function to perform. Value must be the name of some other panel or one of the following:

CLEAR

Clears the panel 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 panel. This permits the addition or modification of input network schedules in a work area.

FE

Combines 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 panel.

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 can SAVE to any input network name but the number of stations specified on the EDIT panel for the network must equal the number defined for the network.

Batch keyword: Positional parameter

n/a

Contains a constant value of INWK for input networks.

Limits: Required for batch only.

Batch keyword: Positional parameter

NETWORK

Specifies the input network name for which schedule information is being defined.

Limits: 1 to 8 alphanumeric characters

Batch keyword: NETWORK

SCAL

Specifies the last two characters of the default base calendar ID to use for schedule resolution. The complete calendar ID used is in the format SCAL*nnxx* 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 panel.

Limits: 2 alphanumeric characters

Required for SAVE, REPL, and RESOLV functions.

Optional for CLEAR, DELETE, EDIT, FE, and FETCH functions.

Batch keyword: SCAL

SCHID-COUNT

Identifies a system-generated field that shows how many schedule IDs currently exist for this network.

Input Network Scheduling Parameter Edit Panel

The Input Network Scheduling Parameter Edit panel lets you define or review scheduling parameters for input networks that are to be scheduled on a date and time basis.

To display the panel, enter EDIT as the function on the DB.2.2 panel.

To exit the panel:

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

```
---------------  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: ____________________________________________________
                                                    DEFAULT SCAL: xx
-- _ -- SYMETRIC  START: ___   SPAN: ___                SCHID-COUNT: nnn

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

This panel contains the following fields:

FUNCTION

Specifies the function to perform. Value must be the name of some other panel 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 panel 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 panel of user input data.

LIST

Lists all SCHIDs, one per panel.

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 panel. Work area data (changes) are retained but the database is not updated.

SR

Combination function that results in a SAVE, a return to the DB.2.2 panel, 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 Workload Automation SE database. If not specified, the scheduling changes are lost once leaving this panel. Entering the SR function automatically returns you to the prior scheduling panel.

SS

Combination function that results in a SAVE, a return to the DB.2.2 panel, and a subsequent SAVE to add a newly defined schedule in the database.

Batch keyword: Positional parameter

NWK

Reflects the network name from the previous panel. This is informational only.

SCHID

(Optional) Specifies the numeric schedule ID on which to perform the specified action.

Limits: 1 to 3 numeric characters from 1 to 255

Batch keyword: SCHID

SCAL

(Optional) Overrides the default SCAL for this SCHID. If this SCHID is using the default, this field is blank.

Limits: 2 alphanumeric characters

Batch keyword: SCAL

ROLL

(Optional) Specifies the action to take when a schedule day falls on a base calendar nonavailable processing day. If used, the value must be one of the following:

B

Roll the schedule back to the previous available processing day in the Base Calendar.

F

Roll the schedule forward to the next available processing day.

N

Do not roll. Schedule day must stand.

D

Do not roll and do not schedule.

Default: D

Batch keyword: ROLL

INDEX

(Optional) Specifies an adjustment to schedule days. After exercising the ROLL option, the schedule is adjusted, forward or backward, by the number of working days entered.

Limits: 1 to 4 numeric characters specified as I*nnn* where I can be plus (unsigned) or minus (-) and *nnn* can be 0 to 365 days

Batch keyword: INDEX

STATION

(Optional) Specifies 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 can appear multiple times in the list. This field is for station identification purposes only. For batch, see the note under the TIME batch keyword.

Limits: 1 to 8 alphanumeric characters

Batch keyword: Not applicable (see TIME)

DOTM

Specifies the due-out time of day for the network workstation using this schedule ID.

Limits: 1 to 4 numeric characters specified as *hhmm*, where *hh* can be 00 through 24 and *mm* can be 00 through 59

Required for ADD or REPL functions.

Optional for DELETE, EXIT, FORMAT, LIST, SAVE, SR, or SS functions.

Batch keyword: TIME

Note for batch input: The following format must be used, where the numeric value (or *n*) corresponds to the relative number of the station.

```
TIME=(dotm1,ldtm1,dody1,dotm2,ldtm2,dody2,...,dotmn,ldtmn,dodyn)
```

LDTM

Specifies 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.

Limits: 1 to 4 numeric characters specified as *hhmm*, where *hh* can be 00 through 24 and *mm* can be 00 through 59

Required for ADD or REPL functions.

Optional for DELETE, EXIT, FORMAT, LIST, SAVE, SR, or SS functions.

Batch keyword: TIME

DODY

Specifies 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.

Default: 0

Limits: 1 to 4 numeric characters from 0 through 255

Required if the lead time (elapsed time) between workstations in a network causes the schedule to cross any midnight boundary.

Batch keyword: TIME

DAILY

(Optional) Specifies to define a daily schedule. Daily means every available processing day as defined by the Base Calendar. Enter an X or Y to select this.

Batch keyword: DAILY

WEEKLY

(Optional) Specifies that the user wants to define a weekly schedule. Enter an X or Y to select this. If WEEKLY is used, the run days of the week must be selected using the SUN through SAT fields.

Batch keyword: WEEKLY

SUN thru SAT

These fields are used to define specific days of the week on which the network is to be run.

Online:

Place an X in panel field names SUN through SAT to indicate the weekly run days.

Limits: Required if WEEKLY is used.

Batch keyword: DAYS

Batch:

Place an X in any of the seven 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

(Optional) Specifies that the user wants to define a monthly schedule. If this field is used, the user can 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. Enter an X or Y to select this.

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

(Optional) Online:
Place an X in panel 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

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

(Optional) 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.

Limits: 1 to 14 numeric characters

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

Specifies 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 wanted run day just as it appears on the panel following the weekly field (for example, SUN, MON, and so on).

Online:

DAY-OF-WEEK:TUE,THU
(separated by blanks or commas)

Limits: 3 to 27 alphanumeric characters

Required if WEEK is used.

Batch keyword: DOW

Batch:

DOW=TUE THU (separated by blanks)

RDAY

(Optional) Specifies 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 you have days of the month when this network is not to run, a slash (/) can be used with the unsigned or negative values. Valid positive values range from 1 to 31. Valid negative values range from -0 to -30.

Limits: 1 to 60 numeric characters

Batch keyword: RDAYS

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

(Optional) Specifies an annual schedule. To select this type of schedule, enter an X or Y in this field. If ANNUAL is used, DAY must also be specified.

Batch keyword: ANNUAL

DAY

Specifies on which days of the annual schedule the user wants 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.

Limits: 1 to 55 numeric characters

Required 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

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

SYMETRIC

(Optional) Used with the START and SPAN fields, defines a symmetric schedule. (This field is misspelled intentionally on the panel to provide compatibility with the batch keyword that cannot exceed eight 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.

Batch keyword: SYMETRIC

START

Used with the SYMETRIC field, sets a starting day.

Limits: 1 to 3 numeric characters specified as Julian day from 1 to 365

Required 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

Used with SYMETRIC and START, defines 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.

Limits: 1 to 3 numeric characters no greater than 255

Required if the SYMETRIC option is chosen.

Batch keyword: SPAN

SCHID-COUNT

Identifies a system-generated field containing the current schedule ID count for this network.

Example: Batch Input

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

```
DBM
SCHD
INWK
EDIT,NETWORK=TSTINWK2,SCAL=PE
ADD,SCHID=1,DAILY=Y,TIME=(0800,0100,,0900,0030)
SAVE
SAVE,NETWORK=TSTINWK2,SCAL=PE
```

DB.2.3 - Output Network Scheduling Panel

The DB.2.3 - Output Network Scheduling panel lets you define or review options taken for output network schedules.

To display the panel, enter:

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

To exit the panel:

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

```
----------------------  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:
```

This panel contains the following fields:

FUNCTION

Specifies the function to perform. Value must be the name of some other panel or one of following:

CLEAR

Clears the panel 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 panel. This enables the addition or modification of network schedules in a work area.

FE

Combines 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 panel.

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 can SAVE to any output network name but the number of stations in the EDIT panel for the network must equal the number defined for the network.

Batch keyword: Positional parameter

n/a

Contains a constant value of ONWK for output networks.

Limits: Required for batch only.

Batch keyword: Positional parameter

NETWORK

Specifies the output network name for which schedule information is being defined.

Limits: 1 to 8 alphanumeric characters

Batch keyword: NETWORK

SCHID-COUNT

Identifies a system-generated field containing the user the current schedule ID count for this network.

Note: Output networks have no SCAL field because their schedule is actually based on the schedule of a connected job. Also, the RESOLV function has no effect on output network schedules.

Usage Notes

If the network is connected to the job with a DB.3.4 function, the network is scheduled automatically whenever the connected job is scheduled. However, the network must be scheduled with DB.2.3 so that time values can be given.

Optionally, for networks that direct the manual handling of SYSOUT type data, users of CA Dispatch can have that product automatically issue a DMDNW command for the network. In this case, the network must not be connected to a job with the DB.3.4 panel.

Output Network Scheduling Parameter Edit Panel

The Output Network Scheduling Parameter Edit panel lets you 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.

To display the panel, enter EDIT as the function on the DB.2.3 panel.

To exit the panel:

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

```
-------------  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:
```

This panel contains the following fields:

FUNCTION

Specifies the function to perform. Value must be the name of some other panel 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 panel, restores the SCHID-COUNT and schedule data to that which existed when EDIT was initially entered and clears the work area.

FORMAT

Clears the panel of user input data.

LIST

Lists all existing SCHIDs, one per panel.

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 panel. Work area data (changes) are retained but the database is not updated.

SR

Combination function that results in a SAVE, a return to the DB.2.3 panel, 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 Workload Automation SE database. If not specified, the scheduling changes are lost once leaving this panel. Entering the SR function automatically returns you to the prior scheduling panel.

SS

Combination function that results in a SAVE, a return to the DB.2.3 panel, and a subsequent SAVE to add a newly defined schedule in the database.

Batch keyword: Positional parameter

NWK

Reflects the network name from the previous panel. This is informational only.

SCHID

Specifies the numeric schedule ID on which the user wants to perform the specified action.

Limits: 1 to 3 numeric characters from 1 through 255

Batch keyword: SCHID

STATION 1 thru 9

(Optional) Specifies 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 can appear multiple times in the list. This field is for station identification purposes only.

Limits: 1 to 8 alphanumeric characters

Batch keyword: Not applicable (see TIME)

DOTM

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

Limits: 4 numeric characters specified as *hhmm*, where *hh* can be 00 through 24 and *mm* can be 00 through 59

Required for ADD or REPL functions.

Optional for DELETE, EXIT, FORMAT, LIST, SAVE, SR, and SS functions.

Batch keyword: TIME

Note for batch input: The following format must be used, where the numeric value (or *n*) corresponds to the number of the station.

```
TIME=(dotm1,ldtm1,dody1,dotm2,ldtm2,dody2,...,dotmn,ldtmn,dodyn)
```

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 panel definition matches the job name to which the network is connected, 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 stations in the network is calculated as follows:

Station# 1

x+*y*(1)

Station# 2

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

Station# 3

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

--- ---

x=Job's DOTM+network connection LDTM+LDTM(1)

y=DODY*2400 to calculate the day displacement.

LDTM

Specifies 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.

Limits: 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

Specifies 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.

Default: 0

Limits: 1 to 4 numeric characters from 0 through 255

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

Identifies a system-generated field containing the current schedule ID count for this network.

Example: Batch Input

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

```
DBM
SCHD
ONWK
EDIT,NETWORK=TSTONWK2
ADD,SCHID=1,TIME=(0800,0100,,0900,0030)
SAVE
SAVE,NETWORK=TSTONWK2
```

Trigger Scheduling Panels

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

Note: All trigger commands share common field requirements.

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 can be triggered per triggering element, or one job can be triggered by up to 255 elements.

More information:

DB.2.6 - Data Set Triggering Panel (see page 131)

DB.2.4 - Job Triggering Panel

The DB.2.4 - Job Triggering panel lets you review or define jobs that trigger scheduling of other jobs when they successfully complete.

To display the panel, enter:

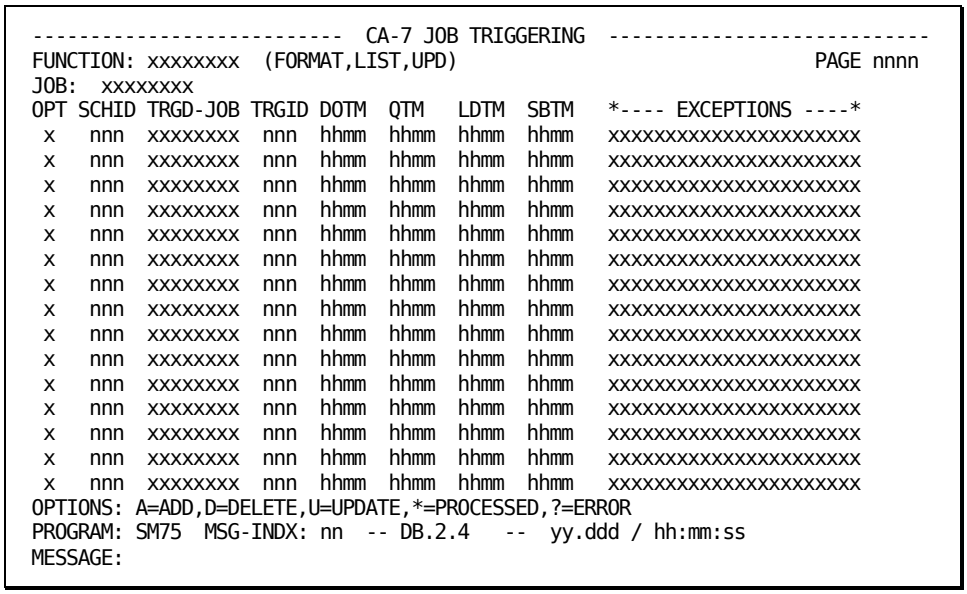
- 4 as the FUNCTION on the DB.2 Menu panel.
- 2.4 as the FUNCTION on the DB Menu panel.
- DB.2.4 as the FUNCTION on any other menu or formatted input panel.
- DB.2.4 or SCHD,JTRG as a top line command.

To exit the panel:

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

Use the JOB field to define the triggering job.

Other keyword descriptions are the same as the DB.2.6 – Data Set Triggering panel.



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

DB.2.5 - Input Network Triggering Panel

The DB.2.5 - Input Network Triggering panel lets you define or review input networks that trigger job scheduling when their last workstation is logged out.

To display the panel, enter:

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

To exit the panel:

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

Use the NWK, DSNBR, or both keyword fields to define the triggering input network.

Other keyword descriptions are the same as DB.2.6 – Data Set Triggering Panel.

```
----------------------  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  hhmm  hhmm  hhmm   xxxxxxxxxxxxxxxxxxxxx
 x   nnn  xxxxxxxx  nnn  hhmm  hhmm  hhmm  hhmm   xxxxxxxxxxxxxxxxxxxxx
 x   nnn  xxxxxxxx  nnn  hhmm  hhmm  hhmm  hhmm   xxxxxxxxxxxxxxxxxxxxx
 x   nnn  xxxxxxxx  nnn  hhmm  hhmm  hhmm  hhmm   xxxxxxxxxxxxxxxxxxxxx
 x   nnn  xxxxxxxx  nnn  hhmm  hhmm  hhmm  hhmm   xxxxxxxxxxxxxxxxxxxxx
 x   nnn  xxxxxxxx  nnn  hhmm  hhmm  hhmm  hhmm   xxxxxxxxxxxxxxxxxxxxx
 x   nnn  xxxxxxxx  nnn  hhmm  hhmm  hhmm  hhmm   xxxxxxxxxxxxxxxxxxxxx
 x   nnn  xxxxxxxx  nnn  hhmm  hhmm  hhmm  hhmm   xxxxxxxxxxxxxxxxxxxxx
 x   nnn  xxxxxxxx  nnn  hhmm  hhmm  hhmm  hhmm   xxxxxxxxxxxxxxxxxxxxx
 x   nnn  xxxxxxxx  nnn  hhmm  hhmm  hhmm  hhmm   xxxxxxxxxxxxxxxxxxxxx
 x   nnn  xxxxxxxx  nnn  hhmm  hhmm  hhmm  hhmm   xxxxxxxxxxxxxxxxxxxxx
 x   nnn  xxxxxxxx  nnn  hhmm  hhmm  hhmm  hhmm   xxxxxxxxxxxxxxxxxxxxx
 x   nnn  xxxxxxxx  nnn  hhmm  hhmm  hhmm  hhmm   xxxxxxxxxxxxxxxxxxxxx
 x   nnn  xxxxxxxx  nnn  hhmm  hhmm  hhmm  hhmm   xxxxxxxxxxxxxxxxxxxxx
 x   nnn  xxxxxxxx  nnn  hhmm  hhmm  hhmm  hhmm   xxxxxxxxxxxxxxxxxxxxx
OPTIONS: A=ADD,D=DELETE,U=UPDATE,*=PROCESSED,?=ERROR
PROGRAM: SM75  MSG-INDX: nn  -- DB.2.5   --  yy.ddd / hh:mm:ss
MESSAGE:
```

DB.2.6 - Data Set Triggering Panel

The DB.2.6 - Data Set Triggering panel lets you define or review data sets that trigger job scheduling when either their creating jobs complete successfully or when the created/updated data set is closed. This function is related to the POST AT CLOSE TIME field on the DB.6 – Data Set Definition Panel.

To display the panel, enter:

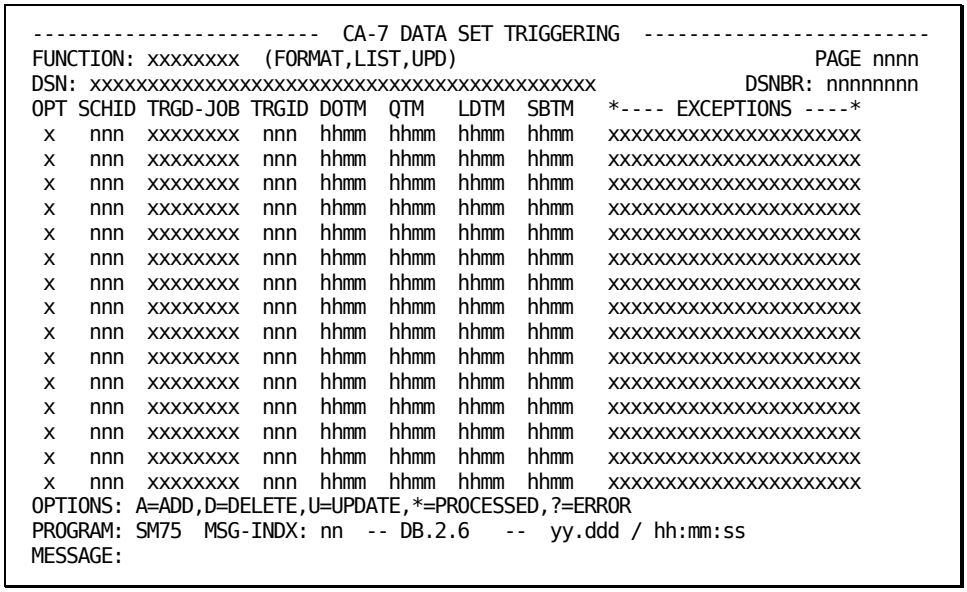
- 6 as the FUNCTION on the DB.2 Menu panel.
- 2.6 as the FUNCTION on the DB Menu panel.
- DB.2.6 as the FUNCTION on any other menu or formatted input panel.
- DB.2.6 or SCHD,DTRG as a top line command.

To exit the panel:

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

Use the DSN, the DSNBR, or both keywords to define the triggering data set.

Partitioned data sets (PDS) cannot be used as data set triggers.



```
------------------------  CA-7 DATA SET TRIGGERING  -------------------------
FUNCTION: xxxxxxxx  (FORMAT,LIST,UPD)                              PAGE nnnn
DSN: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx          DSNBR: nnnnnnnn
OPT SCHID TRGD-JOB TRGID DOTM  QTM   LDTM  SBTM   *---- EXCEPTIONS ----*
 x   nnn  xxxxxxxx  nnn  hhmm  hhmm  hhmm  hhmm   xxxxxxxxxxxxxxxxxxxxx
 x   nnn  xxxxxxxx  nnn  hhmm  hhmm  hhmm  hhmm   xxxxxxxxxxxxxxxxxxxxx
 x   nnn  xxxxxxxx  nnn  hhmm  hhmm  hhmm  hhmm   xxxxxxxxxxxxxxxxxxxxx
 x   nnn  xxxxxxxx  nnn  hhmm  hhmm  hhmm  hhmm   xxxxxxxxxxxxxxxxxxxxx
 x   nnn  xxxxxxxx  nnn  hhmm  hhmm  hhmm  hhmm   xxxxxxxxxxxxxxxxxxxxx
 x   nnn  xxxxxxxx  nnn  hhmm  hhmm  hhmm  hhmm   xxxxxxxxxxxxxxxxxxxxx
 x   nnn  xxxxxxxx  nnn  hhmm  hhmm  hhmm  hhmm   xxxxxxxxxxxxxxxxxxxxx
 x   nnn  xxxxxxxx  nnn  hhmm  hhmm  hhmm  hhmm   xxxxxxxxxxxxxxxxxxxxx
 x   nnn  xxxxxxxx  nnn  hhmm  hhmm  hhmm  hhmm   xxxxxxxxxxxxxxxxxxxxx
 x   nnn  xxxxxxxx  nnn  hhmm  hhmm  hhmm  hhmm   xxxxxxxxxxxxxxxxxxxxx
 x   nnn  xxxxxxxx  nnn  hhmm  hhmm  hhmm  hhmm   xxxxxxxxxxxxxxxxxxxxx
 x   nnn  xxxxxxxx  nnn  hhmm  hhmm  hhmm  hhmm   xxxxxxxxxxxxxxxxxxxxx
 x   nnn  xxxxxxxx  nnn  hhmm  hhmm  hhmm  hhmm   xxxxxxxxxxxxxxxxxxxxx
 x   nnn  xxxxxxxx  nnn  hhmm  hhmm  hhmm  hhmm   xxxxxxxxxxxxxxxxxxxxx
 x   nnn  xxxxxxxx  nnn  hhmm  hhmm  hhmm  hhmm   xxxxxxxxxxxxxxxxxxxxx
 x   nnn  xxxxxxxx  nnn  hhmm  hhmm  hhmm  hhmm   xxxxxxxxxxxxxxxxxxxxx
OPTIONS: A=ADD,D=DELETE,U=UPDATE,*=PROCESSED,?=ERROR
PROGRAM: SM75  MSG-INDX: nn  -- DB.2.6   --  yy.ddd / hh:mm:ss
MESSAGE:
```

This panel contains the following fields:

FUNCTION

Specifies the function to perform. Value must be the name of some other panel or one of the following:

FORMAT

Clears the panel 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.

Batch keyword: Positional parameter

n/a

Contains a constant value depending on the triggering element, either JTRG, DTRG, or NTRG for job, data set, or network respectively.

Limits: Required for batch only.

Batch keyword: Positional parameter

JOB

Specifies the job name whose successful completion causes triggering.

Limits: 1 to 8 alphanumeric characters and required for job triggers.

Batch keyword: JOB

DSN

Specifies the data set name whose creation/update causes triggering of a job.

Note: For VSAM files, the cluster name should be specified.

Limits: 1 to 44 alphanumeric characters and required for data set triggers unless DSNBR is entered.

Batch keyword: DSN

NWK

Specifies the input network name whose completion causes triggering.

Limits: 1 to 8 alphanumeric characters and required for network triggers unless DSNBR is entered.

Batch keyword: NWK

DSNBR

Specifies the existing database member whose creation/completion causes triggering. It is the value assigned to the database member by CA Workload Automation SE.

Limits: 1 to 6 numeric characters for network triggers
1 to 8 numeric characters for data set triggers

Required for data set or network triggers unless DSN/NWK is entered.

Batch keyword: DSNBR

OPT

(Optional) Used with the UPD function, denotes the operation to perform. Valid codes are A (add), D (delete), and U (update).

Batch keyword: OPT

SCHID

(Optional) Specifies for which schedule ID of the triggering element the TRGD-JOB is to be scheduled.

Default: 0 (all schedule IDs)

Limits: 1 to 3 numeric characters from 0 to 255. This field is not valid with OPT=U.

Batch keyword: SCHID

Note: For the data set trigger panel where the data set is one that is tracked with use of SASSXDSN (externally tracked data sets), the SCHID field must be 000 for the trigger to work.

TRGD-JOB

Specifies the job name that is to be triggered by the completion/creation of the triggering element.

Limits: 1 to 8 alphanumeric characters. This field is not valid with OPT=U.

Batch keyword: TJOB

TRGID

(Optional) Denotes a replacement schedule ID to use when the TRGD-JOB is placed into the request queue.

Default: 0 (no replacement)

Limits: 1 to 3 numeric characters from 0 through 255

Batch keyword: TRGID

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.

Example:

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 that this occurrence of JOBX will not trigger JOBY again.

DOTM

Specifies the due-out time of day of TRGD-JOB rounded down to ten minute increments. If this field is used, QTM must be omitted.

Note: If used and the triggering job's DOTM is after the triggered job's DOTM, CA Workload Automation SE assumes the following calendar day.

Limits: 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. If specified, the lowest value is 10.

Required only if QTM omitted.

Batch keyword: DOTM

QTM

Specifies the elapsed queue time of TRGD-JOB rounded down to ten minute increments. If this field is used, DOTM must be omitted because 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.

Limits: 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

(Optional) Specifies the elapsed lead time for TRGD-JOB rounded to ten minute increments. This field specifies the lead or processing time necessary to ensure that TRGD-JOB meets its due-out time.

Default: 0000

Limits: 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.

Batch keyword: LEADTM

SBTM

(Optional) 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.

Default: 0 (no specific submit time requirement)

Limits: 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.

Batch keyword: SBTM

EXCEPTIONS

Indicates an informational only field. Brief messages appear here when an exception occurs trying to process the entry on this line.

Batch keyword: Not applicable

Messages that 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 Workload Automation SE specialist for assistance.

INVALID VALUE *fieldname*

The value for the indicated field is not valid.

NO ROOM IN DATABASE

The CA Workload Automation SE database is full. Contact your installation's CA Workload Automation SE specialist for assistance.

SCHID 0 ALREADY EXISTS

Trigger is already defined for schedule ID zero that 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.

More information:

DB.6 - Data Set Definition Panel (see page 299)

Example: Batch Input

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

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

DB.2.7 - Modification to Resolved Schedule Dates Panel

The DB.2.7 - Modification to Resolved Schedule Dates panel lets you, with its rapid and direct method, permanently alter 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.

To display the panel, enter:

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

To exit the panel:

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

```
---------------  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    LAST RESOLV: yy.ddd / hh:mm
         ....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:
```

This panel contains the following fields:

Note: DB.2.7 has *no batch keywords* because this is an *online only* function.

FUNCTION

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

FORMAT

Clears the panel of user input data.

LIST

Lists schedule information from the database.

UPD

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

JOB

(Optional) Defines the job name for which schedule information is to be listed or altered.

Limits: 1 to 8 alphanumeric characters. If this field is used, NETWORK must be omitted.

NETWORK

(Optional) Defines the name of an input network for which schedule information is to be listed or altered.

Limits: 1 to 8 alphanumeric characters. If this field is used, JOB must be omitted.

MODSTAT

(Optional) Specifies the status of modifications made to the schedule. The following are the possible values:

Blanks

Specifies that no changes have been made to this schedule because the last RESOLV was done.

CURRENT

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

OVERLAID

Specifies that at least one modification had been made previously with this panel, but a subsequent RESOLV for the schedule overlaid the modification that had been made without changing the schedule.

Default: Blanks

SCHID

Specifies which schedule ID of the job or network is to be listed or updated.

Limits: 1 to 3 numeric characters

YEAR

Specifies the calendar year for the first month of the schedule displayed.

LAST RESOLV

Specifies the date and time of the last RESOLV of the schedule displayed.

JAN thru DEC

(Optional) Specifies the run and nonrun Julian days of the job or network for the month indicated. Any scheduled day (1) can 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.

1

Specifies days on which work is done, a scheduled day.

0

Specifies days on which work is not done, a nonscheduled day.

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.

Usage Notes

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

Once displayed, you can change the values as necessary to accomplish the wanted changes.

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

DB.2.8 - Base Calendar Maintenance Panel

The DB.2.8 - Base Calendar Maintenance panel lets you list, add, update, and delete the base calendars. New and updated calendars are immediately available to other functions, such as Resolve, without the need for a SCAL statement in the initialization file.

To use the online base calendar maintenance, define the calendar PDS to CA Workload Automation SE through the CALENDAR statement in the initialization file.

Note: For more information about the CALENDAR statement, see the *Systems Programming Guide*.

To display the panel, enter:

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

To exit the panel:

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

```
----------------------  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         BEGIN      END
     JAN  nnnnnnnnnnnnnnnnnnnnnnnnnnnnnnn         01 / 01   01 / 31
     FEB  nnnnnnnnnnnnnnnnnnnnnnnnnnnn            02 / 01   02 / 28
     MAR  nnnnnnnnnnnnnnnnnnnnnnnnnnnnnnn         03 / 01   03 / 31
     APR  nnnnnnnnnnnnnnnnnnnnnnnnnnnnn           04 / 01   04 / 30
     MAY  nnnnnnnnnnnnnnnnnnnnnnnnnnnnnnn         05 / 01   05 / 31
     JUN  nnnnnnnnnnnnnnnnnnnnnnnnnnnnn           06 / 01   06 / 30
     JUL  nnnnnnnnnnnnnnnnnnnnnnnnnnnnnnn         07 / 01   07 / 31
     AUG  nnnnnnnnnnnnnnnnnnnnnnnnnnnnnnn         08 / 01   08 / 31
     SEP  nnnnnnnnnnnnnnnnnnnnnnnnnnnnn           09 / 01   09 / 30
     OCT  nnnnnnnnnnnnnnnnnnnnnnnnnnnnnnn         10 / 01   10 / 31
     NOV  nnnnnnnnnnnnnnnnnnnnnnnnnnnnn           11 / 01   11 / 30
     DEC  nnnnnnnnnnnnnnnnnnnnnnnnnnnnnnn         12 / 01   12 / 31

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

This panel contains the following fields:

Note: DB.2.8 has *no batch keywords* because this function is an *online only* function.

FUNCTION

Specifies the function to perform. Value must be the name of some other panel or one of the following:

FORMAT

Clears the panel of user input data.

LIST

Lists data for the calendar that is specified in the CALENDAR field.

ADD

Adds the calendar that is displayed. The YEAR field is required if the calendar name is not in standard SCAL*yyxx* format.

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 is used. The YEAR field is required if the calendar name is not in standard SCAL*yyxx* format.

DELETE

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

REFRESH

Fetches the load module copy of the calendar that is specified in the CALENDAR field and uses it to replace the copy in the CA Workload Automation SE calendar PDS.

CALENDAR

Specifies the name of the CA Workload Automation SE calendar you want to act on. For a calendar to be used by CA Workload Automation SE to resolve job/network schedules, the name must be in the format SCAL*yyxx*; where, *yy* is the last two digits of the year it represents and *xx* is a two-position suffix to identify the calendar uniquely. All calendar names that begin with SCAL must adhere to this format. However, you can create and save calendars under other names to use as models or templates for the creation of other calendars.

Limits: 1 to 8 alphanumeric characters

SCHONLY

(Optional) If Y (YES) is specified, CA Workload Automation SE counts only available processing days when resolving a schedule with a number of days relative to the beginning or end of the month (RDAY). The RDAY is set on the DB.2.1 panel.

If SCHONLY is set to N (NO), CA Workload Automation SE counts all days when calculating days relative to the beginning or end of the month.

Default: N (NO)

YEAR

Specifies the year that the calendar represents. When using the standard CA Workload Automation SE calendar naming convention (SCAL*yyxx*), 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 is interpreted as 1998). Years 00 through 74 are considered to be part of the twenty-first century (that is, 02 is interpreted as 2002).

Limits: 2 numeric characters

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

DAYS OF THE MONTH

Represents each month 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

Specifies the beginning and ending days of that month for each month. Use this field when the calendar being defined has nonstandard months. For example, a fiscal or accounting calendar. A year-end boundary can only be crossed in the first or twelfth month.

Required: Yes - for ADD or UPD functions

Usage Notes

- DB.2.8 is only available as an online function.
- To use online base calendar maintenance, the CA Workload Automation SE calendar PDS must be defined to CA Workload Automation SE through the CALENDAR statement in the CA Workload Automation SE initialization file (see the *Systems Programming* Guide).
- Base calendars are automatically added to the CA Workload Automation SE calendar PDS when:
  - A CALBLK statement is included in the CA Workload Automation SE initialization file and the load module copy exists but the PDS has no copy.
  - A reference is made to a base calendar and the PDS has no copy, the load module does not exist, but the corresponding PCALYY*xx* member does exist in the PCDSN.
  - An ADD function is performed from the online base calendar maintenance facility.
- If you have assembled and link edited a new copy of a calendar and want to propagate it to the CA Workload Automation SE calendar PDS, use the REFRESH function. This replaces the existing copy in the CA Workload Automation SE calendar PDS with a new converted copy of the load library version.
- If you have modified the perpetual calendar criteria of an existing calendar, and want to regenerate the calendar with the new criteria, use the REFRESH function. This function replaces the current calendar with a calendar generated from the updated criteria. Be aware that if an assembled and link edited version of this calendar exists in the load library, the assembled and link edited version is used.
- Base calendars that reside in the CA Workload Automation SE calendar PDS do not require a CALBLK statement in the CA Workload Automation SE initialization file.
- 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 Workload Automation SE calendar PDS must have valid PDS member names (one to eight characters starting with an alpha or national character).
- The DELETE function removes the specified calendar from the CA Workload Automation SE calendar PDS ONLY. It has no effect on a load module copy of the same calendar.
- To list all calendar names in the CA Workload Automation SE calendar PDS, issue the following topline command:

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

Chapter 4: Defining Requirements

This section describes how CA Workload Automation SE determines requirements from JCL, how to define and maintain job requirements, and how requirements are satisfied.

This section contains the following topics:

Execution Requirements and the LOAD Process (see page 145)
Satisfy Requirements (see page 148)
Predecessor and Successor Definition (see page 149)

Execution Requirements and the LOAD Process

This topic discusses dependence definition, database definitions, data set dependencies, other dependencies, and temporary predecessors.

Dependence Definition

As work is defined in the CA Workload Automation SE 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 Workload Automation SE can have predecessor requirements. CA Workload Automation SE 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 Workload Automation SE queues

Note: Jobs should not have more than 255 requirements.

Database Definitions

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

Also discussed are defining those output networks that must be invoked upon normal completion of a job and defining report IDs that a job can produce.

Data Set Dependencies

Note: The LOAD process is not used with cross-platform jobs, because that job type does not reference any data sets and has no steps. Thus, there should be no data set dependencies for cross-platform jobs. Data set requirements can be used when a cross-platform job must wait for a data set before being submitted for execution.

The LOAD process addresses the requirements that are 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 panel, it does not run before any required data set is created.

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

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 panel functions. The LOAD process does not delete any requirements that were manually updated or added through a DB.3 panel. This means that data set name changes require DB.3 updates to delete outdated references unless RENAME was used on the DB.6 panel before LOAD occurred.

Other Dependencies

For a CPU job, the DB.1 panel contains fields such as HOLD, VERIFY, and JCL-OVRD. With these fields, 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 panel lets you dynamically add some of these same predecessor requirements to jobs on a temporary basis.

For internal cross-platform jobs, the DB.10 and DB.A.*x* panels contain the fields HOLD and VERIFY. These fields permit the dynamic definition of predecessor requirements once the job is in the request queue.

A job's JCL or PARM data can also include, on a scheduled basis if wanted, 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 that are known as scheduled overrides. Based on how these overrides are scheduled, they can be permanent or temporary without having been defined to the database.

More information:

Scheduled Overrides (see page 311)

Temporary Predecessors

The predecessor requirements are typically defined on a permanent basis. However, occasionally it is necessary on a one-time basis to add new ones or skip existing ones temporarily.

The DB.3 panels enable the temporary requirements with the NEXT-RUN facility. Any such one-time 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 panels reflect the letter O to identify a NEXT-RUN=ONLY predecessor. The letter A identifies one added through the ADDRQ top line command. NEXT-RUN=SKIP predecessors do not appear on the QM.2 update panels as they are skipped when the job enters the queue.

Satisfy Requirements

The system examines all of a job's defined predecessor requirements each time the job is scheduled. The system 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 panels as MCNT. When the count reaches zero, the job moves to the ready queue and then is available for execution.

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

- Successful job completion of a predecessor job under the control of CA Workload Automation SE
- Input network completion of a network connection

  When the last workstation in an input network is logged complete by the workstation operator, CA Workload Automation SE posts this network requirement as complete for any dependent jobs.
- Data set availability posting by the following items:
  - SMF feedback
  - Batch card load program (BLCP)
  - Trailer step
  - U7SVC
- System date/time specifications

Most outstanding requirements needing manual intervention are related to the following items:

- User requirements
- Manually prepared overrides
- Manual verifications
- Restart requirements
- Hold requirements

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

- SUBTM
- VERIFY

The QM.2 panel 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 panel.

If the requirements for a job are predictable, they can lend themselves to use of special trailer step job steps.

Note: For more information about external communicators, see the *Interface Reference Guide*.

Effect of Time Zone Normalization

As stated in the previous topic, the successful job completion of a predecessor job under the control of CA Workload Automation SE is one of the requirements that the system automatically tracks and satisfies.

Without time zone normalization, or if you specify TZPREDS=EXEC on the SMF initialization statement, the start and end times used to calculate this initial requirement are the time zone where the job executed and the time zone where CA Workload Automation SE is running.

With time zone normalization, if you specify TZPREDS=CA7 on the SMF initialization statement, the job's start and end times are normalized to the time zone where CA Workload Automation SE is running.

For example, you have job JOBAAA, and you are specifying predecessor job JOBPPP. When you specify job JOBPPP using panel DB.3.2, you can also enter lead time for that job. This lead time (also known as look back time) specifies that since the last run of job JOBAAA, JOBPPP has started and completed successfully within the last *nn* hours. Values for *nn* range from 1 to 98.

For this example, the current time in New York is 1730 and in California it is 1430. JOBAAA last ran in New York at 1615. The LEADTM specified for JOBPPP is 2. The previous run of JOBPPP ran in California and ended at 1215.

Without time zone normalization, the difference between 1615 and 1215 is four hours and the initial requirement is not satisfied. With time zone normalization, the normalized end time for JOBPPP is now 1515 (New York time), and the difference is only one hour. The initial requirement is satisfied.

Predecessor and Successor Definition

Use the DB.3*n* panels in the following topics to define predecessors and successors.

DB.3 - Job Predecessor/Successor Menu Panel

The DB.3 - Job Predecessor/Successor Menu panel lets you select various panels for defining *connections* (predecessor requirements) to jobs.

To display the panel, enter:

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

To exit the panel:

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

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

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


EXECUTION REQUIREMENTS DEFINED BY:
   1 - DATA SET PREDECESSORS
   2 - CPU JOB PREDECESSORS  OR
       MUTUALLY EXCLUSIVE JOBS (CAN NOT RUN AT SAME TIME)
   4 - INPUT NETWORK PREDECESSORS  OR
       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:
```

DB.3.1 - Data Set Predecessors Panel

For CPU jobs, data set requirements are initially set by CA Workload Automation SE at initial job LOAD time. In other words, when a job runs for the first time, CA Workload Automation SE 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 - Data Set Predecessors panel lets you modify such requirements or to add data set requirements for data sets that are not used as input for the job.

To display the panel, enter:

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

To exit the panel:

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

```
------------------------  CA-7 DATA SET PREDECESSORS  ------------------------
FUNCTION: xxxxxxxx  (FORMAT,LIST,UPD)                                PAGE nnnn
PRED FOR JOB: xxxxxxxx                    LIST-SCHID: nnn                NEXT
OPT SCHID LEADTM *-------------- DATASET NAME --------------*  DSNBR PERM -RUN
 x   nnn   hhhh  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx nnnnnnn  x  xxxx
 x   nnn   hhhh  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx nnnnnnn  x  xxxx
 x   nnn   hhhh  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx nnnnnnn  x  xxxx
 x   nnn   hhhh  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx nnnnnnn  x  xxxx
 x   nnn   hhhh  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx nnnnnnn  x  xxxx
 x   nnn   hhhh  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx nnnnnnn  x  xxxx
 x   nnn   hhhh  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx nnnnnnn  x  xxxx
 x   nnn   hhhh  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx nnnnnnn  x  xxxx
 x   nnn   hhhh  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx nnnnnnn  x  xxxx
 x   nnn   hhhh  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx nnnnnnn  x  xxxx
 x   nnn   hhhh  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx nnnnnnn  x  xxxx
 x   nnn   hhhh  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx nnnnnnn  x  xxxx
 x   nnn   hhhh  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx nnnnnnn  x  xxxx
 x   nnn   hhhh  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx nnnnnnn  x  xxxx
 x   nnn   hhhh  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx nnnnnnn  x  xxxx
 x   nnn   hhhh  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx nnnnnnn  x  xxxx
OPTIONS: A=ADD,D=DELETE,U=UPDATE,*=PROCESSED,?=ERROR
PROGRAM: SM61  MSG-INDX: nn  -- DB.3.1   --  yy.ddd / hh:mm:ss
MESSAGE:
```

This panel contains the following fields:

FUNCTION

Specifies the function to perform. Value must be the name of some other panel or one of the following:

FORMAT

Clears the panel of user input data.

LIST

Lists data from the database. In batch, a formatted panel is not listed; only a found or not found message is returned.

UPD

Updates a data set requirement according to OPT field values.

Batch keyword: Positional parameter

n/a

Predecessor type. Contains a constant value of DSN for data sets.

Limits: Required for batch only.

Batch keyword: Positional parameter

PRED FOR JOB

Specifies the required job name for which data set predecessors are being defined.

Limits: 1 to 8 alphanumeric characters

Batch keyword: Positional parameter

LIST-SCHID

(Optional) 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.

Default: Null (Causes connections for all schedule IDs to be listed)

Limits: 1 to 3 numeric characters

Batch keyword: Not applicable

OPT

(Optional) Used with the UPD function, denotes 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.

Batch keyword: OPT

SCHID

(Optional) Specifies 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 Workload Automation SE. (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 cannot be changed with OPT=U.

Default: 0 (requirement holds for all schedule IDs)

Limits: 1 to 3 numeric characters from 0 through 255

Batch keyword: SCHID

LEADTM

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

0

Specifies that the data set must have been created or updated since the start time of the last run of the job. The amount of time that may have elapsed has no special consideration.

99

Specifies that 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

Specifies that the data set must have been created or updated within the last *nn* hours. Values for *nn* can be 1 to 98.

Default: 0

Batch keyword: LEADTM

DATASET NAME

Defines the fully qualified data set name that is to be defined as a data set requirement for the job in the JOB field.

Limits: 1 to 44 alphanumeric characters and required unless DSNBR is used.

Batch keyword: DSN

DSNBR

Specifies the data set number (as previously assigned by CA Workload Automation SE) that is to be defined as a data set requirement for the job in the JOB field.

Limits: 1 to 8 numeric characters and required unless DSNAME is used.

Batch keyword: DSNBR

PERM

(Optional) Specifies whether this data set is to be considered permanent by CA Workload Automation SE for this job's requirements (Y or N). The DB.6 panel can be used to mark a data set as permanent for all jobs. This field defines the data set as permanent only for this job.

Default: N

Batch keyword: PERM

NEXT-RUN

(Optional) Specifies 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.

YES

Specifies normal, recurring predecessor.

ONLY

Specifies one-time predecessor for only the next run of this job. Automatically discarded by CA Workload Automation SE 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

Specifies normal, recurring predecessor that 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)

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.

Usage Notes

You can add more requirements on a one-time basis. You can skip existing requirements on a one-time basis.

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

- The data set is TYPE=PERM on DB.6 panel.
- The requirement is marked PERM for this job on the DB.3.1 panel.
- The job is marked MAINT=Y on the DB.1 panel.
- The requirement does not apply to this run (schedule ID dependency).
- The requirement was defined as NEXT-RUN=SKIP.
- A RUN or LOAD command scheduled the job.
- 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 LOADDSNS 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) panel. The result is that there are no data set connections for these jobs unless added they are added manually.

Example: Batch Input

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
```

DB.3.2 - Job Predecessors Panel

The DB.3.2 - Job Predecessors panel lets you establish predecessor job requirements.

To display the panel, enter:

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

To exit the panel:

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

```
------------------------  CA-7 CPU JOB PREDECESSORS  ------------------------
FUNCTION: xxxxxxxx  (FORMAT,LIST,UPD)                              PAGE nnnn
PRED FOR JOB: xxxxxxxx                      LIST-SCHID: nnn
OPT SCHID LEADTM    PRED-JOB  NEXT-RUN
 x   nnn   hhhh     xxxxxxxx    xxxx
 x   nnn   hhhh     xxxxxxxx    xxxx
 x   nnn   hhhh     xxxxxxxx    xxxx
 x   nnn   hhhh     xxxxxxxx    xxxx
 x   nnn   hhhh     xxxxxxxx    xxxx
 x   nnn   hhhh     xxxxxxxx    xxxx
 x   nnn   hhhh     xxxxxxxx    xxxx
 x   nnn   hhhh     xxxxxxxx    xxxx
 x   nnn   hhhh     xxxxxxxx    xxxx
 x   nnn   hhhh     xxxxxxxx    xxxx
 x   nnn   hhhh     xxxxxxxx    xxxx
 x   nnn   hhhh     xxxxxxxx    xxxx
 x   nnn   hhhh     xxxxxxxx    xxxx
 x   nnn   hhhh     xxxxxxxx    xxxx
 x   nnn   hhhh     xxxxxxxx    xxxx
OPTIONS: A=ADD,D=DELETE,U=UPDATE,*=PROCESSED,?=ERROR
PROGRAM: SM61  MSG-INDX: nn  -- DB.3.2   --  yy.ddd / hh:mm:ss
MESSAGE:
```

This panel contains the following fields:

FUNCTION

Specifies the function to perform. The value must be the name of some other panel or one of the following:

FORMAT

Clears the panel of user input data.

LIST

Lists data from the database. In batch, a formatted panel 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.

Batch keyword: Positional parameter

n/a

Predecessor type. Contains a constant value of JOB for jobs.

Limits: Required for batch only.

Batch keyword: Positional parameter

PRED FOR JOB

Specifies the job name for which job requirements are being defined or listed.

Limits: 1 to 8 alphanumeric characters

Batch keyword: Positional parameter

LIST-SCHID

(Optional) 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.

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

Limits: 1 to 3 numeric characters

Batch keyword: Not applicable

OPT

(Optional) Used with the UPD function, denotes 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.

Batch keyword: OPT

SCHID

(Optional) Specifies 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 Workload Automation SE. (For the online LIST function, the field on the same line as JOB field defaults to null causing all schedule IDs to be listed.)

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

Limits: 1 to 3 numeric characters from 0 through 255

Batch keyword: SCHID

LEADTM

(Optional) Specifies satisfaction lead time in hours. If nonzero, this value overrides any satisfaction lead time indicated on the DB.1 panel for this job connection. Values are the following:

0

Specifies that the predecessor job must have started and completed successfully since the last run of the dependent job. The amount of time that may have elapsed has no special consideration.

99

Specifies that 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

Specifies that since the last run of this job, the predecessor job has started and completed successfully within the last *nn* hours. Values for *nn* can be 1 to 98.

Default: 0

Batch keyword: LEADTM

PRED-JOB

Names the job on which the job specified in the JOB field is dependent. This predecessor job name can 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 are 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 Workload Automation SE produces a warning message indicating that PRED-JOB is not found.) Conditional dependiencies cannot be specified as generic.

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

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 multiple occurrences are 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

(Optional) Specifies 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

Specifies normal, recurring predecessor.

ONLY

Specifies one-time predecessor for only the next run of this job. Automatically discarded by CA Workload Automation SE 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

Specifies normal, recurring predecessor that 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)

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.

Usage Notes

You can add more requirements on a one-time basis. You can skip existing requirements on a one-time basis.

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

- A RUN or LOAD command scheduled the job.
- 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.

Example: Batch Input

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

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

DB.3.4 - Input/Output Network Tasks Panel

The DB.3.4 - Input/Output Network Tasks panel lets you define a job's requirements (or connections) for workstation networks. One network can be connected to multiple jobs.

To display the panel, enter:

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

To exit the panel:

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

```
--------------------  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    xxxxxxxx  xxxxxxxx    nnn     xxxxxxxx      xxxx
 x   nnn   nnnn    xxxxxxxx  xxxxxxxx    nnn     xxxxxxxx      xxxx
 x   nnn   nnnn    xxxxxxxx  xxxxxxxx    nnn     xxxxxxxx      xxxx
 x   nnn   nnnn    xxxxxxxx  xxxxxxxx    nnn     xxxxxxxx      xxxx
 x   nnn   nnnn    xxxxxxxx  xxxxxxxx    nnn     xxxxxxxx      xxxx
 x   nnn   nnnn    xxxxxxxx  xxxxxxxx    nnn     xxxxxxxx      xxxx
 x   nnn   nnnn    xxxxxxxx  xxxxxxxx    nnn     xxxxxxxx      xxxx
 x   nnn   nnnn    xxxxxxxx  xxxxxxxx    nnn     xxxxxxxx      xxxx
 x   nnn   nnnn    xxxxxxxx  xxxxxxxx    nnn     xxxxxxxx      xxxx
 x   nnn   nnnn    xxxxxxxx  xxxxxxxx    nnn     xxxxxxxx      xxxx
 x   nnn   nnnn    xxxxxxxx  xxxxxxxx    nnn     xxxxxxxx      xxxx
 x   nnn   nnnn    xxxxxxxx  xxxxxxxx    nnn     xxxxxxxx      xxxx
 x   nnn   nnnn    xxxxxxxx  xxxxxxxx    nnn     xxxxxxxx      xxxx
 x   nnn   nnnn    xxxxxxxx  xxxxxxxx    nnn     xxxxxxxx      xxxx
OPTIONS: A=ADD,D=DELETE,U=UPDATE,*=PROCESSED,?=ERROR
PROGRAM: SM61  MSG-INDX: nn  -- DB.3.4   --  yy.ddd / hh:mm:ss
MESSAGE:
```

This panel contains the following fields:

FUNCTION

Specifies the function to perform. Value must be the name of some other panel or one of following:

FORMAT

Clears the panel of user input data.

LIST

Lists data from the database. In batch, a formatted panel is not listed; only a found or not found message is returned.

UPD

Updates a network requirement according to OPT field values.

Batch keyword: Positional parameter

n/a

Predecessor type. Contains a constant value of NWK for networks.

Limits: Required for batch only.

Batch keyword: Positional parameter

TASK FOR JOB

Specifies the job name for which a network connection is to be established.

Limits: 1 to 8 alphanumeric characters

Batch keyword: Positional parameter

LIST-SCHID

(Optional) 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.

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

Limits: 1 to 3 numeric characters

Batch keyword: Not applicable

OPT

(Optional) Used with the UPD function operation, denotes the operation codes: A (add), D (delete), and U (update).

Note: The U option applies only to the LEADTM, NWK-SCHID, DESCRIPTION, and NEXT-RUN fields.

Batch keyword: OPT

SCHID

(Optional) Specifies 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 Workload Automation SE.

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

Limits: 1 to 3 numeric characters from 0 through 255

Batch keyword: SCHID

LEADTM

(Optional) For an input network, specifies satisfaction lead time in hours. Values are the following:

0

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

9999

Specifies that 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

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

Default: 0

Batch keyword: LEADTM

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

Default: 0

Limits: 1 to 4 numeric characters specified as *hhmm* where hours are 00 through 24 and minutes are 00 through 59, the highest value being 2400.

Batch keyword: LEADTM

NETWORK

Specifies the network name.

Limits: 1 to 8 alphanumeric characters

Batch keyword: NETWORK

SUB-ID

(Optional) Specifies a name that 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.

Limits: 1 to 8 alphanumeric characters

Batch keyword: SUBID

NWK-SCHID

(Optional) Specifies a network schedule identification. This field specifies 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 panel, the output network is not scheduled.

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

Limits: 1 to 3 numeric characters specified as 0 through 255

Batch keyword: NWKSCHID

DESCRIPTION

(Optional) Provides further identification of the connected network.

Limits: 1 to 8 alphanumeric characters

For output networks, this field shows when listing the network.
For input networks, this field is only documentation.

Batch keyword: DESC

NEXT-RUN

(Optional) Specifies 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

Specifies normal, recurring predecessor.

ONLY

Specifies one-time predecessor for only the next run of this job. Automatically discarded by CA Workload Automation SE 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

Specifies normal, recurring predecessor that 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)

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.

Usage Notes

You can add more requirements on a one-time basis. You can skip existing requirements on a one-time 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:

- A RUN or LOAD command scheduled the job.
- 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 job enters the request queue. After the network is defined with the DB.5 panel, the network is logically connected to one or more jobs with the DB.3.4 panel. (Required scheduling parameters must also be entered with the DB.2.3 panel. The schedule, however, is not RESOLVed.)

Example: Batch Input

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 that are adjusted by one hour. A name of RPT1205 further identifies this network from any other that could be associated with the job.

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

DB.3.6 - User Memo-Form Predecessors Panel

The DB.3.6 - User Memo-Form Predecessors panel lets you enter a user requirement, free-form text, that can be displayed on a CA Workload Automation SE terminal and that requires an operator response before the connected job can execute. The text should direct someone to perform some manual task. Multiple lines may be required to completely describe the action needed. The POST command, QM.2, or QM.1 panels can be used to indicate that the task has been completed (the requirement has been satisfied).

To display the panel, enter:

- 6 as the FUNCTION on the DB.3 Menu panel.
- 3.6 as the FUNCTION on the DB Menu panel.
- DB.3.6 as the FUNCTION on any other menu or formatted input panel.
- DB.3.6 or JOBCONN,USR as a top line command.

To exit the panel:

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

```
---------------------  CA-7 USER MEMO-FORM PREDECESSORS  ----------------------
FUNCTION: xxxxxxxx  (FORMAT,LIST,UPD)                                 PAGE nnnn
PRED FOR JOB: xxxxxxxx                      LIST-SCHID: nnn
OPT SCHID    *--- MEMO-FORM USER PREDECESSOR ---*  NEXT-RUN
 x   nnn     xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx    xxxx
 x   nnn     xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx    xxxx
 x   nnn     xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx    xxxx
 x   nnn     xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx    xxxx
 x   nnn     xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx    xxxx
 x   nnn     xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx    xxxx
 x   nnn     xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx    xxxx
 x   nnn     xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx    xxxx
 x   nnn     xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx    xxxx
 x   nnn     xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx    xxxx
 x   nnn     xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx    xxxx
 x   nnn     xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx    xxxx
 x   nnn     xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx    xxxx
 x   nnn     xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx    xxxx
 x   nnn     xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx    xxxx
OPTIONS: A=ADD,D=DELETE,U=UPDATE,*=PROCESSED,?=ERROR
PROGRAM: SM61  MSG-INDX: nn  -- DB.3.6   --  yy.ddd / hh:mm:ss
MESSAGE:
```

This panel contains the following fields:

FUNCTION

Specifies the function to perform. Value must be the name of some other panel or one of the following:

FORMAT

Clears the panel of user input data.

LIST

Lists data from the database. In batch, a formatted panel is not listed; only a found or not found message is returned.

UPD

Updates a user requirement according to OPT field values.

Batch keyword: Positional parameter

n/a

Predecessor type. Contains a constant value of USR for user text.

Limits: Required for batch only.

Batch keyword: Positional parameter

PRED FOR JOB

Specifies the job name for which user requirements are being defined.

Limits: 1 to 8 alphanumeric characters

Batch keyword: Positional parameter

LIST-SCHID

(Optional) 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.

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

Limits: 1 to 3 numeric characters

Batch keyword: Not applicable

OPT

(Optional) Used with the UPD function, denotes 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).

Batch keyword: OPT

SCHID

(Optional) Specifies 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 Workload Automation SE.

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

Limits: 1 to 3 numeric characters from 0 through 255

Batch keyword: SCHID

MEMO-FORM USER PREDECESSOR

Specifies a free-form description of the user requirement. Not valid with OPT=U. Any text can 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 that 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.

Limits: 1 to 36 alphanumeric characters

Batch keyword: USR

NEXT-RUN

(Optional) Specifies 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

Specifies normal, recurring predecessor.

ONLY

Specifies one-time predecessor for only the next run of this job. Automatically discarded by CA Workload Automation SE 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

Specifies normal, recurring predecessor that 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)

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.

Usage Notes

You can add more requirements on a one-time basis. You can skip existing requirements on a one-time basis.

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

Example: Batch Input

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
```

DB.3.7 - Report IDs Created Panel

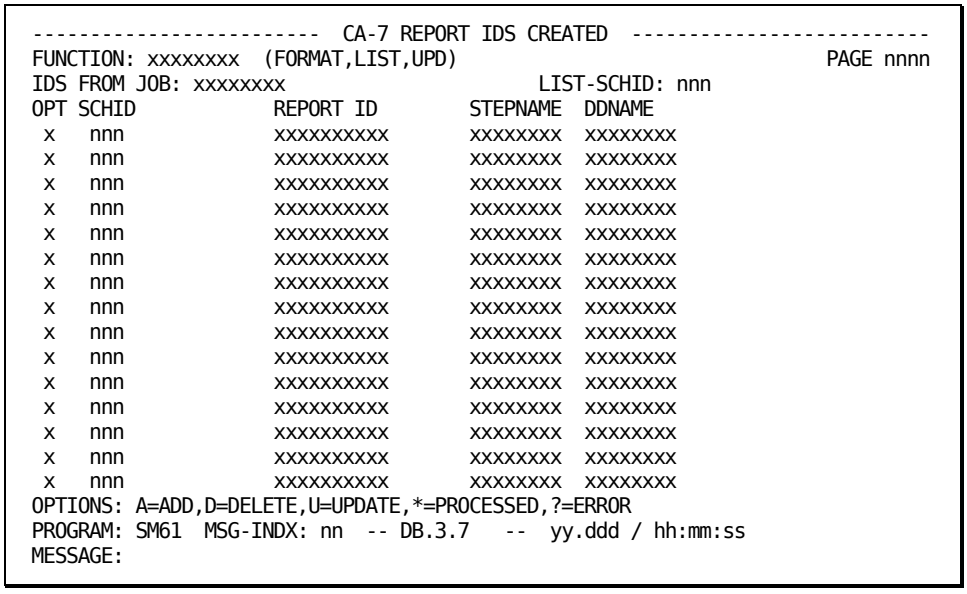
The DB.3.7 - Report IDs Created panel lets you record information in the database regarding reports that CPU jobs produce. The information is optional and serves only as reference material for anyone wanting to review the information at a terminal. Other means of recording this data, including DB.4 functions, may be more desirable.

To display the panel, enter:

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

To exit the panel:

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



```
------------------------  CA-7 REPORT IDS CREATED  -------------------------
FUNCTION: xxxxxxxx  (FORMAT,LIST,UPD)                                PAGE nnnn
IDS FROM JOB: xxxxxxxx                      LIST-SCHID: nnn
OPT SCHID            REPORT ID        STEPNAME  DDNAME
 x   nnn             xxxxxxxxxx       xxxxxxxx  xxxxxxxx
 x   nnn             xxxxxxxxxx       xxxxxxxx  xxxxxxxx
 x   nnn             xxxxxxxxxx       xxxxxxxx  xxxxxxxx
 x   nnn             xxxxxxxxxx       xxxxxxxx  xxxxxxxx
 x   nnn             xxxxxxxxxx       xxxxxxxx  xxxxxxxx
 x   nnn             xxxxxxxxxx       xxxxxxxx  xxxxxxxx
 x   nnn             xxxxxxxxxx       xxxxxxxx  xxxxxxxx
 x   nnn             xxxxxxxxxx       xxxxxxxx  xxxxxxxx
 x   nnn             xxxxxxxxxx       xxxxxxxx  xxxxxxxx
 x   nnn             xxxxxxxxxx       xxxxxxxx  xxxxxxxx
 x   nnn             xxxxxxxxxx       xxxxxxxx  xxxxxxxx
 x   nnn             xxxxxxxxxx       xxxxxxxx  xxxxxxxx
 x   nnn             xxxxxxxxxx       xxxxxxxx  xxxxxxxx
 x   nnn             xxxxxxxxxx       xxxxxxxx  xxxxxxxx
 x   nnn             xxxxxxxxxx       xxxxxxxx  xxxxxxxx
OPTIONS: A=ADD,D=DELETE,U=UPDATE,*=PROCESSED,?=ERROR
PROGRAM: SM61  MSG-INDX: nn  -- DB.3.7   --  yy.ddd / hh:mm:ss
MESSAGE:
```

This panel contains the following fields:

FUNCTION

Specifies the function to perform. Value must be the name of some other panel or one of the following:

FORMAT

Clears the panel of input data.

LIST

Lists data from the database. In batch, a formatted panel is not listed; only a found or not found message is returned.

UPD

Updates report IDs according to OPT field values.

Batch keyword: Positional parameter

n/a

Predecessor type. Contains a constant value of RPT for reports.

Limits: Required for batch only.

Batch keyword: Positional parameter

IDS FROM JOB

Specifies the job name for which report IDs are being defined.

Limits: 1 to 8 alphanumeric characters

Batch keyword: Positional parameter

LIST-SCHID

(Optional) 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.

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

Limits: 1 to 3 numeric characters

Batch keyword: Not applicable

OPT

(Optional) Used with the UPD function, denotes 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).

Batch keyword: OPT

SCHID

(Optional) Specifies 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 Workload Automation SE. Is not valid with OPT=U.

Default: 0 (specifies the report IDs apply to all schedule IDs)

Limits: 1 to 3 numeric characters from 0 through 255

Batch keyword: SCHID

REPORT ID

(Optional) Specifies the free-form ID of the report.

Limits: 1 to 10 alphanumeric characters

Batch keyword: RPTID

STEPNAME

(Optional) Specifies the JCL step name in which the reports are created.

Limits: 1 to 8 alphanumeric characters

Batch keyword: STEP

DDNAME

(Optional) Specifies the JCL ddname with which the reports are created.

Limits: 1 to 8 alphanumeric characters

Batch keyword: DD

Example: Batch Input

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 the resource usage by using tables internal to CA Workload Automation SE.

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

The VRM resources are ignored for nonexecutable jobs.

This section contains the following topics:

Resource Types (see page 173)
Resource Conflicts Resolution (see page 179)
RM - Virtual Resource Management Menu Panel (see page 179)
RM.1 - Job Resource Management Panel (see page 180)
RM.2 - Job/Resource Cross-Reference List Panel (see page 187)
RM.3 - Active Job Resources Display Panel (see page 190)
RM.4 - Pending Resources Job Display Panel (see page 193)
RM.5 - Jobs Waiting on Resources Panel (see page 196)
RM.6 - Corequisite Resources List Panel (see page 198)
RM.7 - Resource Count Resource Management Panel (see page 200)
VRM Device Control (see page 203)

Resource Types

The following topics discuss the various resource types available and the VRM variable definitions.

Shared Resources

At times, multiple jobs can run using the same resource. You can define this situation as a Shared resource requirement under the Virtual Resource Management facility. A Shared resource connection indicates that multiple jobs can execute simultaneously while using this resource.

Exclusive Resource

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

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 Workload Automation SE 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@

Address Space Resources

The CA Workload Automation SE 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 Workload Automation SE 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 Workload Automation SE ready queue with a status of W-RSRC until the address space becomes active. This check occurs during every CA Workload Automation SE 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 Workload Automation SE is executing.

Resource Count Resources

A resource count resource is a resource that 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. The value identifies the total number of occurrences for this resource that the job uses. 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. Next, suppose that you want 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 uses.

Note: The following resource count resource name prefix is reserved:

- VRMDD.

VRM Variable Definitions

Note: VRM Variable Definitions are specific to CPU jobs because these jobs are submitted to the z/OS operating system where the IBM feature WorkLoad Management (WLM) is active. Internal cross-platform jobs do not use VRM Variable Definitions.

CA Workload Automation SE supports the definition of variables using the RM panels. These variables are known as VRM variables because the definitions are stored in the VRM database.

Currently, the sole purpose of a VRM variable definition is to control the SCHENV keyword insertion. This function is part of the CA Workload Automation SE interface with the IBM Workload Manager.

Note: For more information about this interface, see the *Interface Reference Guide*.

The SCHENV keyword on the JOB statement names the WLM scheduling environment that must be available for the job to run. CA Workload Automation SE supports the dynamic insertion of the SCHENV keyword on the JOB statement when the job is submitted so that manual JCL changes are not required.

The value of SCHENV to be inserted can be defined for a specific SCHID of the job using a VRM variable definition. When CA Workload Automation SE prepares to submit the job, it scans VRM variable definitions for the SCHID of the job. If an appropriate VRM variable definition is found, it is used to set the value of the SCHENV keyword to be inserted.

The Structure of VRM Variable Definitions

You can tailor the SCHENV keyword insertion for a specific run of a job by defining a VRM variable in the resource profile of the job. A VRM variable is indicated when VAR is specified as the resource type in the definition on the RM.1 panel. When VAR is specified, the FREE value is not used and is always set to N. The resource name in the definition must begin with 'SCHENV.'. This prefix signifies that the variable is used for the SCHENV insertion. The second node of the resource name designates the WLM scheduling environment that must be available for the job to run. This value must conform to IBM naming conventions for scheduling environments.

How VRM Variable Definitions Are Used

The CA Workload Automation SE IBM WLM Interface uses VRM variable definitions to control the SCHENV keyword insertion.

The setting of the WLMSE keyword on the OPTIONS statement in the CA Workload Automation SE initialization file governs SCHENV keyword insertion. If N or NO is coded, CA Workload Automation SE does not attempt the SCHENV keyword insertion. WLMSE=Y or WLMSE=YES indicates that CA Workload Automation SE is to insert the SCHENV keyword. If a value other than these values is coded, CA Workload Automation SE assumes that it is a global default. The global default is inserted when CA Workload Automation SE is unable to determine a more specific value for the job.

CA Workload Automation SE scans first for an entry with a SCHID value matching the value of the job being submitted. If a match occurs, the value from the resource name is inserted on the JOB statement. If no match occurs, CA Workload Automation SE scans for an entry with a SCHID value of 0. If found, CA Workload Automation SE uses it as the SCHENV value on the JOB statement. If no variable definition with a SCHID matching the SCHID of the job being submitted is present and if no definition with a SCHID value of 0 is present, CA Workload Automation SE attempts to use the global value that is defined in the initialization file if one is available. If no global value has been defined on the WLMSE keyword of the OPTIONS statement, the job is submitted without any SCHENV keyword.

Examples

In Examples 1 through 3, consider the following variables that are defined for job A:

```
OPT  SCHID   RESOURCE NAME                STEPNAME     TYPE FREE

     0       SCHENV.SHIFT1                             VAR  N
     2       SCHENV.SHIFT2                             VAR  N
     15      SCHENV.SPECIAL                            VAR  N
     20      SCHENV.NONE                               VAR  N
```

Example 1

WLMSE=YES is coded on the OPTIONS statement in the initialization file. If job A is demanded with SCHID=2, CA Workload Automation SE inserts SCHENV='SHIFT2' on the JOB statement for A.

Example 2

WLMSE=YES is coded on the OPTIONS statement in the initialization file. Suppose that job A is demanded with SCHID=3. Because you have no specific definition for SCHID=3, the value from the definition for SCHID=0 is used (SCHENV.SHIFT1).

Example 3

WLMSE=YES is coded on the OPTIONS statement in the initialization file. Schedule scan brings in job A with SCHID=20. 'SCHENV.NONE' is a reserved resource name in a VRM variable definition. The name indicates that the job is submitted without inserting the SCHENV keyword.

In Examples 4 and 5, consider the following variables that are defined for job B. No variable has been defined for SCHID 0.

```
OPT  SCHID   RESOURCE NAME                STEPNAME     TYPE FREE

     5       SCHENV.FIVE                               VAR  N
     55      SCHENV.FIFTY_FIVE                         VAR  N
     100     SCHENV.NONE                               VAR  N
```

Example 4

WLMSE=VANILLA is coded on the OPTIONS statement in the initialization file. Job B is demanded with SCHID=30. SCHID 30 has no specific definition. Since SCHID 0 has no definition, CA Workload Automation SE uses the global value and inserts SCHENV='VANILLA' on the JOB statement for B.

Example 5

WLMSE=YES is coded on the OPTIONS statement in the initialization file. Job B is demanded with SCHID=20. Because SCHID 20 has no specific definition, no generic definition for SCHID 0 and no global default that is specified for WLMSE, the job is submitted without any SCHENV keyword.

How VRM Variable Definitions Differ from Other VRM Definitions

A VRM resource definition usually implies a condition that must be satisfied before the job can be submitted. For example, a definition using the resource type CRQ means that the corequisite resource must be in the state indicated by the FREE value in the definition or CA Workload Automation SE does not submit the job.

But a VRM variable definition does not imply such a condition. Before submitting a job, CA Workload Automation SE tests each of the resource conditions that are specified in the VRM profile of the job. When all these conditions are satisfied, the job is eligible for submission. VRM variable definitions are ignored in this process.

Resource Conflicts Resolution

VRM dynamically manages resource conflicts based on VRM database definitions and CA Workload Automation SE job queue activity. This usually requires little or no manual intervention. CA Workload Automation SE 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 can occur if you have 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) can 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.

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.

Note: For information about handling these anomalies, see the discussion of the OPTIONS statement keyword JSOP in the *Systems Programming Guide*.

RM - Virtual Resource Management Menu Panel

The Virtual Resource Management Menu panel lets you select the Virtual Resource Management functions.

To display the panel, enter:

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

To exit the panel, enter the name of an online panel as the FUNCTION value or move the cursor to the top line and enter a top line command if some other function is wanted.

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

```
-------------------- 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
```

RM.1 - Job Resource Management Panel

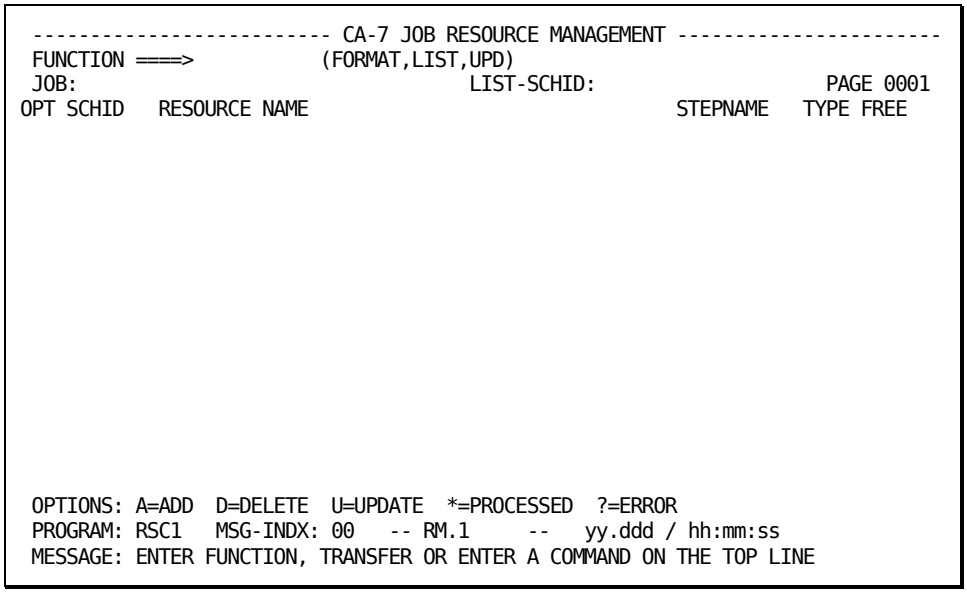
The RM.1 Job Resource Management panel lets you list, add, update, or delete resource connections by job. It generates the VRM static type J and R records maintained on the VRM database component.

To display the panel, enter:

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

To exit the panel:

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



This panel contains the following fields:

FUNCTION

Specifies the function to perform. Value must be the name of some other panel or one of the following:

FORMAT

Clears the panel 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 that follows.

Batch keyword: Positional parameter

JOB

Specifies the job name on which the indicated function is to be performed.

Limits: 1 to 8 alphanumeric characters

Batch keyword: JOB

LIST-SCHID

(Optional) 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.

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

Limits: 1 to 3 numeric characters from 0 to 255

Batch keyword: Not applicable

OPT

Used with the UPD function, denotes the type of connection operation. The operation codes are A (add), U (update), and D (delete).

Batch keyword: OPT

SCHID

(Optional) Specifies 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 Workload Automation SE.

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

Limits: 1 to 3 numeric characters from 0 through 255

Batch keyword: SCHID

RESOURCE NAME

Specifies the resource name being connected to the job.

Limits: 1 to 44 characters. The first character must be nonblank and no embedded blanks.

Exception: If the resource connection is made to a resource count resource, 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

(Optional) Specifies the resource is freed at the conclusion of the named job step. Not valid for types ASX and CRQ.

Limits: 1 to 8 alphanumeric characters

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.

If a job is restarted in a step after this STEPNAME, the resource does not go through free processing until the job completes.

A step flush condition does not post the STEPNAME process.

TYPE

Determines the usage of the resource by the job. The valid resource usage types are:

ASX

Specifies 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 Workload Automation SE is executing.

CRQ

Specifies 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

Specifies an exclusive resource. Only one job can be active with this resource.

RCT

Specifies 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

Specifies a shared resource. Multiple jobs can be active with the same resource.

VAR

Specifies a variable definition. It does not define a necessary resource condition for the job. The FREE value will be forced to N. The resource name for such a resource must conform to the following format:

SCHENV.*x*

x Defines the name of an IBM WLM scheduling environment (1 to 16 characters). This value should follow IBM conventions governing names of scheduling environments.

Batch keyword: TYPE

FREE

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

Specifies the resource is only freed if any step in the job abends (abend or condition code checking).

F

Specifies the resource is 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

Specifies do not free the resource at successful job completion. The resource can be freed with the PRSCF command. A cancel of the job also frees the resource.

Y

Specifies to 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

Specifies the job is submitted only if the resource is active (as established by the PRSQA command). This is the default.

I

Specifies the job is submitted only if the resource is *not* active.

For address space resources, the following FREE options are supported:

A

Specifies the job is submitted only if the resource is active on the same system as CA Workload Automation SE. The resource name is assumed to be a jobname or started task name running on the system. This is the default.

I

Specifies the job is submitted only if the resource is *not* active on the same system as CA Workload Automation SE. The resource name is assumed to be a jobname or started task name running on the system.

For variable resources, the FREE option is not used and is always set to N regardless of any value specified.

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

A

Specifies the resource count is only decremented when the job abends.

F

Specifies the resource count is 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

Specifies the resource count is not decremented when the job completes. The PRSCF command can be used to decrement the resource count. A cancel of the job also decrements the resource count.

Y

Specifies the resource count is 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).

Batch keyword: FREE

Example: Batch Input

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
```

Critical Path Monitoring Corequisite Resources

You can use Critical Path Monitoring with CA Workload Automation SE 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 resource name has the following format:

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

flowname

Defines the 1 to 8 character unique name to assign to this flow.

endjob

Defines the job name of the last job in this flow.

endschid

Defines the schedule ID of the last job in this flow. Use a specific number from 1 to 255 or set to a value of 0 (zero) to indicate the ending job has the same schedule ID as the starting job of the FLOW.

endtarget

Defines the target latest time the ending job should complete (*hhmm*). Alternately, a time interval can be set from the starting time of the FLOW (*+hhmm*).

endday

Defines an optional parameter to add days to the target time (*.n*). You can specify a maximum of 9 days when an ending time (*hhmm*) is used for the *endtarget*. If an interval (*+hhmm*) is used, the maximum *endday* is 8.

Example

```
FLOW@PAYFLOW1.PAYJOBZ.1.1700
```

Usage Notes

The resource type must be corequisite (CRQ).

The resource name must begin with FLOW@.

The flowname, end job name, end schedule ID, and end target are required parameters and must be separated by periods.

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 causes 24 hours to be added to the target time. A value of 2 causes 48 hours to be added, and so forth.

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 want to specify I (inactive) for the FREE type.

Important! VRM resources are never attached to nonexecutable jobs. Thus, the starting job of any flow must be executable.

Note: For more information about the Critical Path Monitoring facility, see the *Interface Reference Guide*

RM.2 - Job/Resource Cross-Reference List Panel

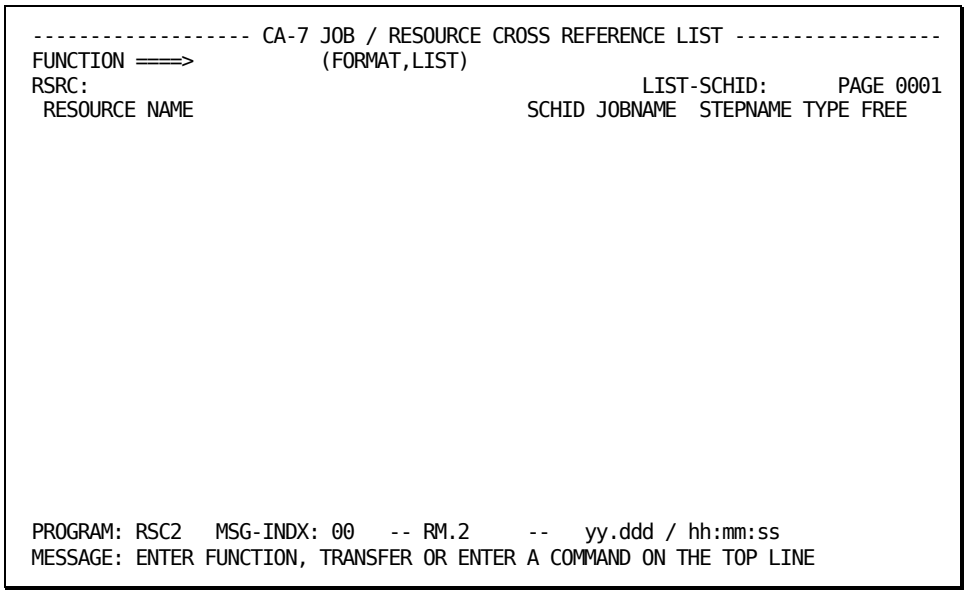
The RM.2 Job/Resource Cross-Reference List panel lets you list jobs using resources. The jobs can be listed by a fully qualified resource name or by a generic key.

To display the panel, enter:

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

To exit the panel:

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



This panel contains the following fields:

FUNCTION

Specifies the function to perform. Value must be the name of some other panel or one of the following:

FORMAT

Clears the panel of user input data.

LIST

Lists resources and the associated jobs based on user specified resource criteria.

Batch keyword: Positional parameter

RSRC

Specifies the resources for which information is to be listed.

*****

Specifies all resources.

resource

Specifies a fully qualified resource name.

Limits: 1 to 44 alphanumeric characters

resource*

Specifies multiple resources specified by a generic resource name.

Limits: 1 to 43 alphanumeric characters terminated with an asterisk

Batch keyword: RSRC

LIST-SCHID

(Optional) 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.

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

Limits: 1 to 3 numeric characters from 0 to 255

Batch keyword: SCHID

RESOURCE NAME

Identifies the requested resources.

SCHID

Identifies the schedule ID of the job using this resource.

JOBNAME

Identifies the name of the job using this resource.

STEPNAME

Identifies the name of the step that frees this resource.

TYPE

Identifies the usage mode of the resource (for example, SHR, EXC).

FREE

Identifies the disposition of this resource at step, abnormal, or successful completion time.

Note: For more information about the FREE keyword, see the RM.1 panel descriptions.

More information:

RM.1 - Job Resource Management Panel (see page 180)

Example: Batch Input

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

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

RM.3 - Active Job Resources Display Panel

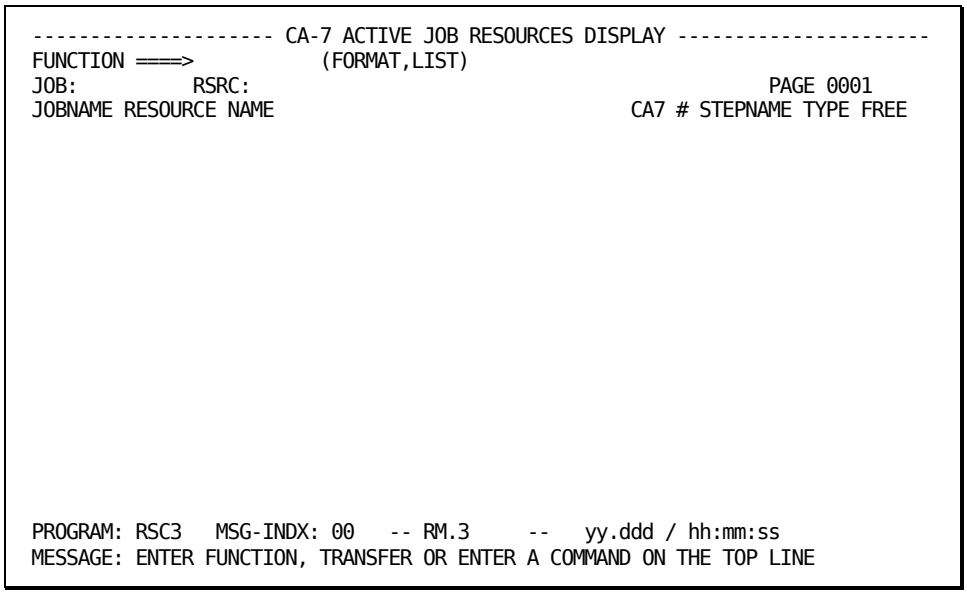
The RM.3 Active Job Resources Display panel lets you display active resources by job. The jobs can be listed generically or with fully qualified job names.

To display the panel, enter:

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

To exit the panel:

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



This panel contains the following fields:

FUNCTION

Specifies the function to perform. Value must be the name of some other panel or one of the following:

FORMAT

Clears the panel of user input data.

LIST

Lists active resources associated with user specified jobs.

Batch keyword: Positional parameter

JOB

(Optional) Specifies a fully qualified or generic job name.

*****

Specifies all jobs.

job

Specifies a fully qualified job name.

Limits: 1 to 8 alphanumeric characters

job*

Specifies multiple jobs specified by a generic job name.

Limits: 1 to 7 alphanumeric characters terminated with an asterisk

Default: All jobs

Batch keyword: JOB

RSRC

(Optional) Specifies a fully qualified or generic resource name. An asterisk indicates a generic request.

*****

Specifies all resources.

resource

Specifies a fully qualified resource name.

Limits: 1 to 44 alphanumeric characters

resource*

Specifies multiple resources specified by a generic resource name.

Limits: 1 to 43 alphanumeric characters terminated with an asterisk

Default: All resources

Batch keyword: RSRC

JOBNAME

Identifies the job name.

RESOURCE NAME

Identifies the names of the active resources.

CA7 #

Identifies the CA Workload Automation SE job number.

STEPNAME

Identifies the name of the job step that freed the resource.

TYPE

Identifies the usage mode of the resource for the job.

FREE

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

Example: Batch Input

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

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

RM.4 - Pending Resources Job Display Panel

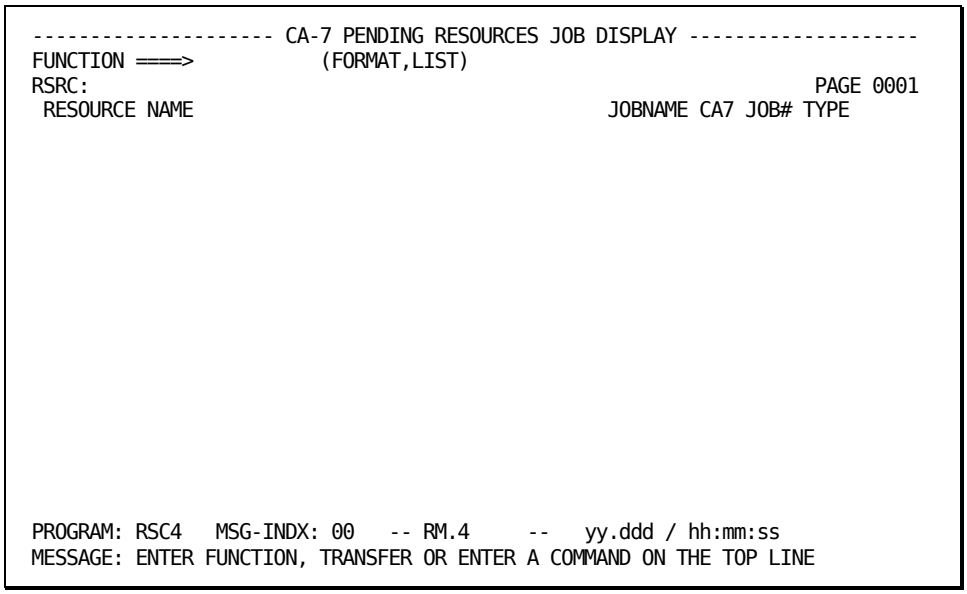
The RM.4 Pending Resources Job Display panel lets you display all pending resources with an associated job name. A pending resource is a nonfreed resource connected to a job that has already executed and has been purged from the request queue. This resource can only be freed by using the PRSCF command. The pending resources can be listed using a generic or fully qualified resource name.

To display the panel, enter:

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

To exit the panel:

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



This panel contains the following fields:

FUNCTION

Specifies the function to perform. Value must be the name of some other panel or one of the following:

FORMAT

Clears the panel of user input data.

LIST

Lists all pending resources (nonfreed) and the associated job name. A pending resource can be freed using the PRSCF command.

Batch keyword: Positional parameter

RSRC

(Optional) Specifies a fully qualified or generic resource name. An asterisk indicates a generic request.

*****

Specifies all resources.

resource

Specifies a fully qualified resource name.

Limits: 1 to 44 alphanumeric characters

resource*

Specifies multiple resources specified by a generic resource name.

Limits: 1 to 43 alphanumeric characters terminated with an asterisk

Default: All resources

Batch keyword: RSRC

RESOURCE NAME

Identifies the pending resource names.

JOBNAME

Identifies the job name.

CA7 JOB#

Identifies the CA Workload Automation SE job number.

TYPE

Identifies the usage mode of the resource by this job.

Example: Batch Input

This batch input example lists all pending VRM resources.

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

RM.5 - Jobs Waiting on Resources Panel

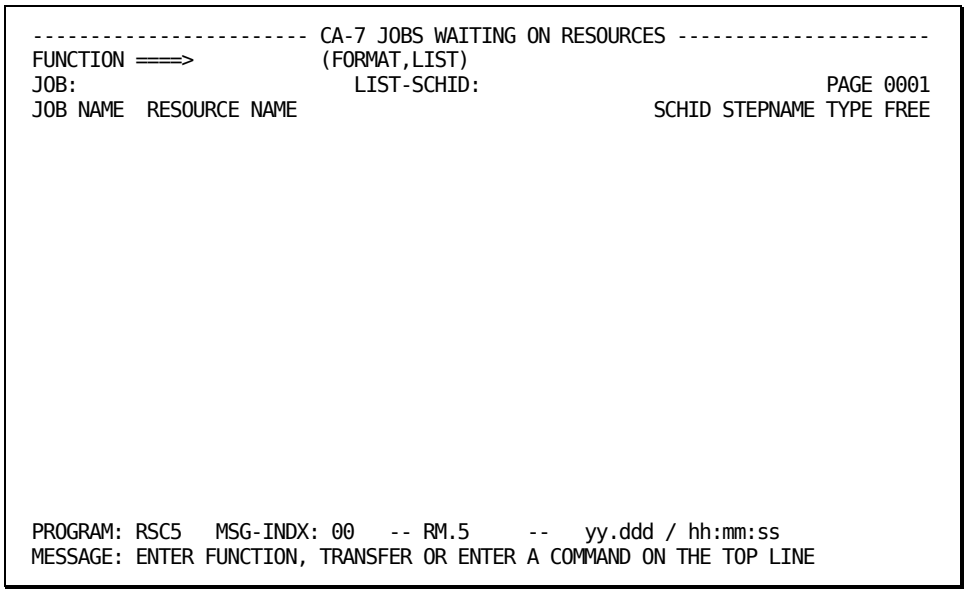
The RM.5 Jobs Waiting on Resources panel lets you display jobs waiting for resources. It can be used to show the resources that a job is waiting for when the job shows a status of W-RSRC on the List Queue (LQ) display. However, ASX resources may not always show on the RM.5 panel. This can occur because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

To display the panel, enter:

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

To exit the panel:

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



This panel contains the following fields:

FUNCTION

Specifies the function to perform. Value must be the name of some other panel or one of the following:

FORMAT

Clears the panel of user input data.

LIST

Lists jobs waiting for resources based on user specified job name and schedule ID criteria.

Batch keyword: Positional parameter

JOB

(Optional) Specifies a fully qualified or generic job name.

***

Specifies all jobs.

job

Specifies a fully qualified job name.

Limits: 1 to 8 alphanumeric characters

job*

Specifies multiple jobs specified by a generic job name.

Limits: 1 to 7 alphanumeric characters terminated with an asterisk

Default: All jobs

Batch keyword: JOB

LIST-SCHID

(Optional) 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.

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

Limits: 1 to 3 numeric characters from 0 to 255

Batch keyword: SCHID

JOB NAME

Identifies the job name of the job awaiting the resource.

RESOURCE NAME

Identifies the name of the nonavailable resource.

SCHID

Identifies the schedule ID of the job awaiting the resource.

STEPNAME

Identifies the jobstep in which the resource is to be freed.

TYPE

Identifies the usage mode of the resource by the job.

Note: On this panel, a TYPE of FRE may show. This would indicate that a PRSCF command with FORCE=YES has been used for this resource.

FREE

Identifies the disposition of the resource at job completion time.

Example: Batch Input

This batch input example lists jobs waiting on VRM resources.

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

RM.6 - Corequisite Resources List Panel

The RM.6 Corequisite Resources List panel lets you list active corequisite resources.

To display the panel, enter:

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

To exit the panel:

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

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









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

This panel contains the following fields:

FUNCTION

Specifies the function to perform. Value must be the name of some other panel or one of the following:

FORMAT

Clears the panel of user input data.

LIST

Lists active corequisite sources.

Batch keyword: Positional parameter

RSRC

Specifies a fully qualified or generic resource name. An asterisk indicates a generic request.

*****

Specifies all resources.

resource

Specifies a fully qualified resource name.

Limits: 1 to 44 alphanumeric characters

resource*

Specifies multiple resources specified by a generic resource name.

Limits: 1 to 43 alphanumeric characters terminated with an asterisk

Batch keyword: RSRC

CO-REQ RESOURCE NAME

Identifies the fully qualified corequisite resource name.

Example: Batch Input

This batch input example lists all active corequisite resources.

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

RM.7 - Resource Count Resource Management Panel

The RM.7 Resource Count Resource Management panel lets you list, add, update, or delete a resource count type resource.

To display the panel, enter:

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

To exit the panel:

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

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

This panel contains the following fields:

FUNCTION

Specifies the function to perform. Value must be the name of some other panel or one of the following:

FORMAT

Clears the panel of all input data.

LIST

Lists the Resource Count Resources and related resource information.

UPD

Used to update information about the Resource Count Resource.

Batch keyword: Positional parameter

RSRC

Specifies a fully qualified or generic Resource Count Resource name. This field is only valid with LIST function.

*****

Specifies all resources.

resource

Specifies a fully qualified resource name.

Limits: 1 to 39 alphanumeric characters

resource*

Specifies multiple resources specified by a generic resource name.

Limits: 1 to 38 alphanumeric characters terminated with an asterisk

Batch keyword: RSRC

OPT

Used with the Update and Add functions, denotes the operation to perform. Valid codes are U (update), A (add), and D (delete).

Batch keyword: OPT

RESOURCE NAME

Identifies the Resource Count Resource for which the option is to be performed.

Batch keyword: DSN

TOTAL COUNT

Identifies the total number of available occurrences for this Resource Count Resource.

Batch keyword: TOTAL

CURR IN USE

Identifies the number of occurrences of this Resource Count Resource currently in use.

Example: Batch Input

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=*
```

VRM Device Control

Note: VRM Device control applies only to CPU jobs, because it reads data obtained from the LOAD process. If you need to control the number of jobs sent from this CA Workload Automation SE Online system to a particular node, you can use the VRM Resource Count function, but you must implement the function manually.

VRM Device Control provides a way to control job submission based on the availability of devices that are detected during the CA Workload Automation SE Database Load process. With this option, CA Workload Automation SE 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 about 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.

VRM Device Control Functions - Definition Structure

CA Workload Automation SE 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.D*xxxxxxxx*.U*yyyyyyyy*/*nnnn*

xxxxxxxx

Specifies the device code in character hex. If the device code is not available, a value of 00000000 is used.

yyyyyyyy

Specifies the UNIT name. If the unit name is not available, no value is used.

nnnn

Specifies the number of references to this device code and UNIT combination calculated during the CA Workload Automation SE Database LOAD process.

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 can be listed on the job's RM.1 panel. The term 'VRMDD.' begins each definition. Because VRM resource names may have already been defined that begin with 'VRMDD.' we recommend using RM.2 to list all VRM resources that conflict before implementation of the feature.

VRM Device Control Functions - Definition Control

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

The Database Load program (SASSJJCL) scans each DD statement in the job and extracts the 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
```

CAL2OPTN includes a SASSDTAB sample. Locate it by finding the entry for VRM Device Control in the AL2$$IDX member.

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 are 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 SASSJJCL. 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.

VRM Device Control Functions - Submission Control

If VRMDD=Y or VRMDD=S is coded on the OPTIONS statement in the CA Workload Automation SE 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 SASSDTAB. For example, after a job is loaded, its RM.1 panel displays the following entries:

```
VRMDD.D3390000F.UDISK/0001
VRMDD.D3338000E.USYSDA/0001
```

When VRM attempts to submit the job, it associates all of the resources according to the groups defined in SASSDTAB. 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.

VRM Device Control - Activation

VRM device control comprises two functions: definition and submission control. Either or both of these functions can be activated depending on installation needs. Activation of these functions is indicated by keyword values on the OPTIONS statement in the CA Workload Automation SE 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

This section contains the following topics:

Overview (see page 209)
Monitor Exception Conditions (see page 210)
Respond to Exception Conditions (see page 211)
Types of Exceptions Recognized by ARF (see page 211)
ARFSET Structure (see page 214)
Implementation Considerations (see page 217)
AR.3 - ARF Condition Definition Maintenance Panel (see page 218)
ARF Condition Definition Edit Panel (see page 220)
Rules for Coding ARF Action Statements (see page 241)
Use Reserved Words in Type Specific Tests (see page 246)
Use Variables in ARF Action Statements (see page 248)
Examples of ARF Condition Definition (see page 249)

Overview

Use the Automated Recovery Facility (ARF) for CA Workload Automation SE 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 the following:

- 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 Workload Automation SE.
- Jobs considered late according to user specified criteria tested when the job begins or completes.

The kinds of recovery actions that can 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 Workload Automation SE or MVS commands.
- Restarting, canceling, or forcing the completion of jobs as the final step in the recovery for job completion exceptions.

Monitor Exception Conditions

Because recovery monitoring can 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 Workload Automation SE 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 can be associated with a job in several ways. An ARFSET can be named on the job definition panel. This can be overridden using the #ARFSET statement in the JCL or PARM data. Both of these designations can be overridden using the ARFSET keyword on the DEMAND command.

For internal cross-platform jobs, each job has only one step. With this step data, the completion information contains only the node name where the job executed, instead of the program name, proc name, or step name. Because jobs can be routed to an alternate node, if defined, these fields may not be reliable in any ARF conditional testing. Consider this impact if defining ARFSETs for internal cross-platform job types.

Respond to Exception Conditions

Recovery actions are scheduled when the exception is detected so that the action can be performed at or near the point of failure. Each occurrence of an exception condition can have up to seven 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 Workload Automation SE 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 Workload Automation SE terminal command or set of such commands. Since all CA Workload Automation SE terminal command input is logged to the CA Workload Automation SE Browse Data Set, all ARF recovery activity is thereby recorded.

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 Workload Automation SE 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 Workload Automation SE is notified of job initiation. If the elapsed time interval specified in the ARF definition expires prior to notification in CA Workload Automation SE of job completion then an ARF exception is recognized. Such monitoring will allow CA Workload Automation SE 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 can 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
- JCL for JCL error conditions
- FL for flush conditions
- HRC for high step condition code or job level cond-code (see the DB.1 - CPU Job Definition panel)

The test also allows a relational operator to be specified along with the value of the code. Thus, for example a test can 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 wildcards and generics is allowed. The following test can 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.

The HRC test is only valid for a JC (Job Completion) exception definition. The highest condition code from any step in the job is used for the HRC test after the job has completed. The HRC test cannot be combined with the other test formats. The STEP, PROC, and PGM applicability tests for a JC definition using HRC must all be EQ *.

Note: We do not recommend a JC (Job Completion) check for a good EOJ condition. A relational operator (RO) of IG on the Job Definition (DB.1) panel assumes the job completes successfully. Any responses specified by a JC ARFSET looking for a good EOJ may not occur.

LA - Late Notification at CA Workload Automation SE Prompting

An ARF exception is recognized when CA Workload Automation SE begins late prompting for the job. The exception occurs only once for the job when it is initially considered late by CA Workload Automation SE. 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 panel 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 Workload Automation SE 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 Workload Automation SE 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 Workload Automation SE 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 can 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 wildcards and generics is allowed. The following test can 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.

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 job definition 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, the job has no ARF monitoring. The ARFSET specification on the job definition panel supplies the ARFSET reference unless it is overridden with the #ARF scheduled override statement in the JCL. Both of these indications can 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 job definition 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 Workload Automation SE job definition also apply to ARFSETs.

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 can be considered to have four parts:

- The filter criteria
- The type-specific tests
- The responses
- The final disposition

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 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 to be monitored by ARF.

Type Specific Tests

If a job passes all of the tests implied by the filter criteria and is monitored by ARF, the type specific tests in the ARF definition are 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.

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.

Types of Responses

Responses are defined using action statements. The format of an action statement is much like that of a CA Workload Automation SE 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.

AC - Issue a Command

The AC action statement is used to issue a CA Workload Automation SE 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 Workload Automation SE 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.

More information:

AR.3 - ARF Condition Definition Maintenance Panel (see page 218)

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 can be restarted, it can be canceled, a normal completion can be forced or if no ARF action is wanted, the job can be left for manual exception handling.

Implementation Considerations

Although every attempt has been made to ensure that resources are efficiently used by ARF, additional resources must be expended to monitor ARF exceptions. The impact of ARF on overall CA Workload Automation SE throughput (if any) will vary depending on the extent of its implementation. We therefore recommend that use of ARF be restricted to automating recovery for jobs with recurrent and predictable problems or for those jobs that are on a *critical path*.

AR.3 - ARF Condition Definition Maintenance Panel

The AR.3 ARF Condition Definition Maintenance panel lets you 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 the panel, enter:

- AR.3 as the function value on any menu or formatted input panel.
- AR.3 as a top line command.

To exit the panel:

- Move the cursor to the top line of the panel and enter the wanted top line command.

This panel contains the following fields:

FUNCTION

Specifies the function to be performed. Value must be the name of another panel or one of the following:

CLEAR

Clears the panel 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

Combines 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.

Batch keyword: Positional parameter

SET NAME

Specifies the required name of the ARFSET that is being defined.

Limits: 1 to 8 alphanumeric characters

Batch keyword: SET

UID

Specifies the required CA Workload Automation SE user security identification.

Limits: 1 to 3 numeric characters from 0 to 255

Batch keyword: UID

RESPONSE ID

Specifies the ID that will be used during ARF recovery. ARF uses an internal CA Workload Automation SE 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.

If CA Workload Automation SE external security SUBCHECK is used, this field is validated to make sure that the user ID updating this AR.3 panel has submit authority for the RESPONSE ID value.

Limits: 1 to 8 alphanumeric characters. The maximum length allowable may vary depending on the external security package used, if any.

Batch keyword: RESPONDR

DEF-COUNT

Specifies a display of the number of definitions that are in the ARF edit work file.

ARF Condition Definition Edit Panel

The ARF Condition Definition Edit panel lets you 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
```

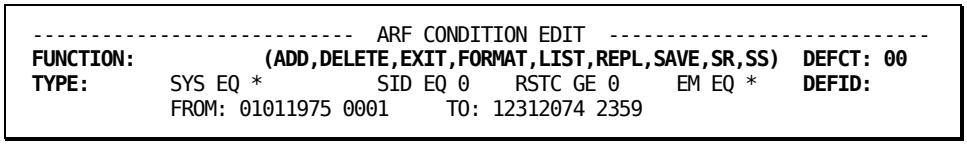
To display the panel, enter EDIT as the function on the AR.3 panel.

To exit the panel:

- Enter EXIT in the function field.
- Move the cursor to the top line of the panel and enter the wanted top line command.

Note: Test format HRC requires STEP, PROC, and PGM set to EQ *.

Field Descriptions



FUNCTION

Specifies the function to be performed. Value must be the name of another panel 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 panel 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 panel input entered by user.

LIST

Lists the definition associated with the current definition index. This function can 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 that results in a SAVE, a return to the AR.3 panel, and a REPL from the AR.3 panel to replace an existing ARFSET on the database.

SS

Combination function that results in a SAVE, a return to the AR.3 panel, and a SAVE from the AR.3 panel to add a new ARFSET to the database.

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 Workload Automation SE initially prompts for a late condition. 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.

Batch keyword: TYPE

DEFID

Identifies an index that uniquely specifies this definition within the ARFSET.

Limits: 1 to 3 decimal characters
Valid values 1-255

Batch keyword: DEFID

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 the job is to be monitored by ARF. Following the literal 'SYS' is a relational operator input field that 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 can be overtyped with an 'NE' to indicate that the system name must not be equal to the value in the following input field.

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 '*' can 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 can incorporate '+' to indicate a wildcard. 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.

Default: *

Limits: 1 to 8 alphanumeric characters
+ indicates a wildcard
* indicates a generic

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 the job is to be monitored by ARF. Following the literal 'SID' is a relational operator input field that 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 can 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.

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' can 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.

Default: 0 (a value of 0 refers to ALL SCHIDs)

Limits: 1 to 3 decimal characters from 0 to 255

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 the job is to be monitored by ARF. Following the literal 'RSTC' is a relational operator input field that 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 can 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.

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' can 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.

Default: 0

Limits: 1 to 3 decimal characters from 0 to 999

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 that 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 can be overtyped with an 'NE' to indicate that the entry mode must not be equal to the value in the following input field.

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 '*' can 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 the following:

DEMD

Job entered through a DEMAND command

DTRG

Job entered through a DSN trigger

JTRG

Job entered through a job trigger

NTRG

Job entered through a network trigger

PSCH

Job entered through personal scheduling

RPET

Job entered because it was repeated

RUN

Job entered through a RUN command

SSCN

Job entered through a schedule scan

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. To pass the test, the queue entry date and time must be greater than the date and time specified here.

FROM Date

Default: 01011975

Limits: 8 numeric character positions in MMDDYYYY format. Valid range from 01011975 to 12312074. You can also use one of the reserved words.

Batch keyword: FROMD

FROM Time

Default: 0001 (1 minute past midnight)

Limits: 4 character positions in *hhmm* format. Valid range from 0000 to 2359.

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

Default: 12312074

Limits: 8 character positions in MMDDYYYY format. Valid range from 01011975 to 12312074. You can also use one of the reserved words.

Batch keyword: TOD

TO Time

Default: 2359 (11:59 PM)

Limits: 4 character positions in *hhmm* format. Valid range from 0000 to 2359.

Batch keyword: TOT

More information:

Use Reserved Words in Type Specific Tests (see page 246)

Field Descriptions - Type Specific Tests for JC and SC Conditions

JC, SC TST

```
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, LB, LE, LS TST:
            RO: GE  DATE:           TIME:           AO: ?  INT/ADJ:
```

HRC test format requires STEP, PROC, and PGM set to EQ *.

Note: We do not recommend a JC (Job Completion) check for a good EOJ condition. A relational operator (RO) of IG on the Job Definition (DB.1) panel assumes the job completes successfully. Any responses specified by a JC ARFSET looking for a good EOJ may not occur.

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 that 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 can be overtyped with an 'NE' to indicate that the step name must not be equal to the value in the following input field.

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 '*' can 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 can incorporate '+' to indicate a wildcard. 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.

Default: *

Limits: 1 to 8 characters, blanks and commas are not allowed
+ indicates a wildcard
* indicates a generic

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 that 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 can be overtyped with an 'NE' to indicate that the proc name must not be equal to the value in the following input field.

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 '*' can 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 can incorporate '+' to indicate a wildcard. 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.

Default: *

Limits: 1 to 8 characters, commas and blanks are not allowed
+ indicates a wildcard
* indicates a generic

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 that 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 can be overtyped with an 'NE' to indicate that the program name must not be equal to the value in the following input field.

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 '*' can 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 can incorporate '+' to indicate a wildcard. 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.

Default: *

Limits: 1 to 8 characters, commas and blanks are not allowed
+ indicates a wildcard
* indicates a generic

Batch keyword: PGM

CC/ABENDS

(Optional) Up to six tests of completion codes can 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. Overtype this field with one of the following values to define a completion code test:

???

Indicates no completion code test.

SYS

Indicates the completion code to be tested is a system abend code.

USR

Indicates the completion code to be tested is a user abend code.

CC

Indicates the completion code to be tested is a condition code.

FL

Indicates to perform a test for a flush condition. If this value is used, the corresponding completion code field is ignored, and the only relational operator arguments permitted are EQ and NE.

JCL

Indicates to perform a test for a JCL error. If this value is used, the corresponding completion code is ignored, and the only relational operator arguments permitted are EQ and NE.

HRC

Indicates to test the highest condition code from any step at the end of the job. An HRC test can only be used in an ARF JC Exception definition. It can only be combined with other HRC tests.

Default: ??? (indicates no test)

Batch keyword: FMT1, FMT2, FMT3, FMT4, FMT5, FMT6

Note: CC tests (in either SC or JC definitions) are performed for each step of a job and may result in multiple ARF responses for the same job. HRC tests are only performed after the last executed step of a job using the highest condition code from any of the executed steps. A positive HRC test results in only one set of responses even if multiple steps meet the test.

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 can 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

Default: GE

Limits: Optional unless the first input has a value other than ???

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 can 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 can contain '+' to indicate a wildcard 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'.

Note: S822 abends cannot be "caught" because the IBM SMF step term record does not contain the needed information.

Default: 000

Limits: For SYS, 3 valid hexadecimal characters
For USR and CC, 1 to 4 decimal characters

Limits: Optional unless the first input has a value other than ???

Batch keyword: VAL1, VAL2, VAL3, VAL4, VAL5, VAL6

Completion code tests can be joined using the following logical connectives:

&

Indicates the tests are to be 'and'ed

/

Indicates the tests are to be 'or'ed

Default: __ indicating no connective

Limits: Optional unless tests are to be joined.

Batch keyword: LO1, LO2, LO3, LO4, LO5

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  __
            ??? 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 can 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 Workload Automation SE attempts to submit the job is compared with the date/time expression 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 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 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 can 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

Default: GE

Limits: Required for EC, IS, LB, LE and LS condition types. It should not be specified for any other condition type.

Batch keyword: TRO

DATE

This input field can 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 Workload Automation SE 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 specified here.

If the condition type is LS, the date/time when the job submits is compared with the date/time expression specified here.

Limits: 8 character positions in MMDDYYYY format. Valid range from 01011975 to 12312074. Or you can use one of the reserved words.

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 can 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 Workload Automation SE 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.

Limits: 4 character positions in *hhmm* format. Valid range from 0000 to 2359. You can also use one of the reserved words.

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 can 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 Workload Automation SE 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.

Limits: 1 position. Allowable values are '+' or '-'.

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 can be used with the following condition types: EC, EE, IS, LB, LE and LS. Data from this field can be interpreted as an adjustment to a date and time already specified; this is its meaning if the condition type is 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.

Limits: 4 character positions in *hhmm* format. Valid range from 0000 to 2359. You can also use one of the reserved words in the table in the Reserved Words in Type Specific Tests.

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

More information:

Use Reserved Words in Type Specific Tests (see page 246)

Field Descriptions - Responses

```
      RESPONSES:

      1:
      2:
      3:
      4:
      5:
      6:
      7:
FINAL --  DISP   : N CA-11?: N BYPGDG: N  USAGE:    PROCESS:    CC:
```

(Optional) In this section of the AR.3 panel are seven lines where you can specify ARF responses. Responses cannot 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.

Limits: 74 character positions for input. Format must conform to rules for coding ARF statements.

Batch keyword: RSP1, RSP2, RSP3, RSP4, RSP5, RSP6, RSP7

Note: Batch syntax requires that action statements be enclosed in parentheses. Additional parentheses can be needed depending on action statement keywords and parameters.

More information:

Rules for Coding ARF Action Statements (see page 241)

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 panel 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 the following:

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 Workload Automation SE top-line RESTART command. Additional parms for the command can 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 Workload Automation SE 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.

Default: N (no action)

Limits: Required if condition type is IS or JC. Ignored for all other condition types.

Batch keyword: DSP

CA-11?

(Optional) This input field indicates whether CA WA Restart Option is to be used for restart. The valid values are the following:

N

No, CA WA Restart Option is not to be used for restart. This is the default.

Y

Yes, CA WA Restart Option is to be used for restart. Other input values in the final disposition section may be relevant.

Default: N (CA WA Restart Option not used)

Limits: Required only if CA WA Restart Option is to be used for restart.

Batch keyword: CA11

BYPGDG

(Optional) This input field indicates the value on the RESTART command. For more information, see the documentation on the RESTART command.

Valid values are the following:

C

CA WA Restart Option to accept GDG bias resolution according to the MVS catalog (CAT option of CA WA Restart Option).

N

This is the default. CA WA Restart Option should NOT bypass GDG logic on restart.

V

CA WA Restart Option to verify that the GDG bias resolution recorded in the CMT agrees with the MVS catalog (VER option of CA WA Restart Option).

Y

Yes, CA WA Restart Option should bypass GDG logic on restart.

Default: N (CA WA Restart Option will not bypass GDG logic)

Batch keyword: BYPGDG

USAGE

(Optional) Supplies the value of the USAGE parameter to be used on the RESTART command. For values, see the CA WA Restart Option documentation. This option honored only if CA WA Restart Option used and CA Workload Automation SE is inserting the RMS step. For more information, see the documentation on the INSERT-RMS field of the DB.1 panel and on the RESTART command.

Limits: 1 alphanumeric character

Batch keyword: USAGE

PROCESS

(Optional) Supplies the value of the PROCESS parameter to be used on the RESTART command. This option honored only if CA WA Restart Option used and CA Workload Automation SE is inserting the RMS step. Valid values are F, P, S, N, O or R. For more information, see the documentation on the INSERT-RMS field of the DB.1 panel and on the RESTART command.

Limits: 1 alphanumeric character

Batch keyword: PROCESS

CC

(Optional) Supplies the value of the CONDCD parameter to be used on the RESTART command. This option honored only if CA WA Restart Option used and CA Workload Automation SE is inserting the RMS step. For more information, see the documentation on the INSERT-RMS field of the DB.1 panel and on the RESTART command.

Limits: 1 to 4 alphanumeric characters from 0-4095

Batch keyword: CC

START

(Optional) Supplies values of the PROCSTRT and STPSTRT parameters on the RESTART command. This option honored only if CA WA Restart Option 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.

For more information, see the documentation on the RESTART command.

Limits: 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.

Batch keyword: START

END

(Optional) Supplies values of the PROCEND and STPEND parameters on the RESTART command. This option honored only if CA WA Restart Option 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.

For more information, see the documentation on the RESTART command.

Limits: 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.

Batch keyword: END

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.

ARF has four 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.

AC - Issue a Command

Statement Identifier

AC

Purpose

The AC action statement is used to issue a CA Workload Automation SE command.

Format

The allowable keyword is:

M=

The data following the M= keyword is assumed to be valid CA Workload Automation SE terminal input. The format of the command must adhere to the conventions used for the TRAILER terminal. The maximum length is 69.

Example

The example illustrates the use of the AC action statement to issue a DEMAND command to request the PAYROLL job.

```
AC,M=DEMAND,JOB=PAYROLL
```

AM - Issue a Message

Statement Identifier

AM

Purpose

Execution of the AM action statement causes a message to be issued to the MVS console or to a specified TSO user.

Format

The following are the allowable keywords:

CM=

The data following the CM= specifies the communications mode for the message. The following are the acceptable values of CM=:

T

Send the message 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 message 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 can be displayed in the next TSO session.

C

Send the message to the MVS console.

H

Send the message to the MVS console and highlight them.

U=

The data following the U= names the TSO user that is to receive the message. 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.

Examples

The example illustrates the use of the AM action statement to send the message "HELLO THERE!" to a TSO user.

```
AM,CM=T,U=TSOUSER1,M=HELLO THERE!
```

The example illustrates the use of the AM action statement to send the message "HELLO CONSOLE!" to the MVS console.

```
AM,CM=C,M=HELLO CONSOLE!
```

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 0001 to 9999. This keyword can also be coded as T=.

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.

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 Workload Automation SE 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 job definition 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 the following:

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 has 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 has no default. Acceptable values are the following:

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.

Example

The example illustrates the use of the AJ action statement to request an ARFJ job named X. If the job does not complete successfully, retry after waiting 1 minute. If the job still does not complete successfully jump to the 4th action statement in the definition.

```
AJ,JOB=X,RETRY=1,DELAY=1,ACTION=4
```

Use 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 cannot 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 panel. 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:

Name	Description	Relevant AR.3 Panel Fields
DOD	The due-out date of the job.	FROM: (first input field) TO: (first input field) DATE:

Name	Description	Relevant AR.3 Panel Fields
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 can 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 four 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.

Use Variables in ARF Action Statements

Certain values in ARF action statements can be referenced using variables. These variables can be used in AM and AC action statements. The following describes each of these variables:

&ARFSET

Specifies the name of the ARFSET referenced by this run of the job.

&AARDATE

Specifies the date the ARFSET referenced by this job was first used for ARF monitoring.

&AARTIME

Specifies the time the ARFSET referenced by this job was first used for ARF monitoring.

&ARFRESP#

Specifies the number of the response being processed.

&ARFSETIX

Specifies the index of the definition being processed.

&INTDATE

Specifies the date this ARF condition was detected.

&INTTIME

Specifies the time this ARF condition was detected.

&JES#

Specifies the JES number of the job in ARF recovery. &JES# is not applicable to cross-platform job types and may not be available for some exception conditions (for example IS, LS, and LA).

&JOB#

Specifies the CA Workload Automation SE job number of the job in ARF recovery.

&JOBNAME

Specifies the name of the job in ARF recovery.

&SCHDID

Specifies the schedule ID of the job in ARF recovery.

&SYSTEM

Specifies the system name of the job in ARF recovery.

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.

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 can be restarted exactly one time.

This example illustrates how CA Workload Automation SE ARF can 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 Workload Automation SE 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 Workload Automation SE 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.

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 Workload Automation SE ARF can 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 that 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!!!
```

No other actions follow this one. Since this is not a job completion event, no final disposition processing occurs. ARF recovery is complete for this exception.

Chapter 7: Workload Documentation

This section contains the following topics:

Overview (see page 253)
Workload Documentation Panels (see page 254)
Special Documentation Features (see page 277)

Overview

CA Workload Automation SE permits the entry of free-form user documentation (prose) of the workload at virtually any level. Prose can range from a general description to specific instructions on how a particular function at a specified workstation is performed. The following items are some of the documentation features:

- Ready access to documentation through CA Workload Automation SE terminals.
- Real-time, online updating from a change control or production control area.
- Segmenting documentation into user-defined categories (that is, restart information, user abends, control statement layouts, balancing instructions, setup requirements, and so forth).
- Adding documentation (prose) through any authorized CA Workload Automation SE terminal using the DB.4 panels and edit facility of database maintenance.

Once defined, you can define documentation:

- With the LPROS command.
- With the DB.4 panels (on which the documentation is also defined and maintained).
- As a list option in several other inquiry commands such as LQ, LPRE, LJOB.
- Automatically in some cases.

More information:

Special Documentation Features (see page 277)

Workload Documentation Panels

CA Workload Automation SE has six predefined major categories of documentation, each with its own unique definition panel. Within the documentation that is defined through these panels, the user can embed special user-defined segments.

You can select the appropriate panel from the DB.4 Menu display. Some separate panels are provided for the following types:

- 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 Workload Automation SE database.

Also, assume that the documentation exists as members of a PDS that is available to CA Workload Automation SE. In this case, the documentation could be fetched and edited using the DB.7 panel and saved in the Active Area. The user could then enter the top line DB.4 command and select the appropriate category. Next, the user can 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.

More information:

Special Documentation Features (see page 277)

DB.4 - Workload Documentation Menu Panel

The DB.4 - Workload Documentation Menu panel lets you select various documentation functions.

To display the panel, enter:

- 4 as the FUNCTION on the DB Menu panel.
- DB.4 as the FUNCTION on any other menu or formatted input panel.
- DB.4 or PROSE as a top line command.

To exit the panel:

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

Select the wanted documentation type by entering the appropriate FUNCTION value and pressing the Enter key.

```
---------------------  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:
```

DB.4.1 - CPU Job Documentation Panel

The DB.4.1 - CPU Job Documentation panel lets you define free-form job level documentation. You can define documentation in one continuous stream for the specified job.

Also, you can subdivide the documentation into segments and subsegments.

To display the panel, enter:

- 1 as the FUNCTION on the DB.4 Menu panel.
- 4.1 as the FUNCTION on the DB Menu panel.
- DB.4.1 as the FUNCTION on any other menu or formatted input panel.
- DB.4.1 or PROSE,JOB as a top line command.

To exit the panel:

- Press PF3 to return to the DB.4 Menu panel except with the TSO/ISPF interface when PF3 has been assigned to another function.
- Move the cursor to the top line of the panel and enter the wanted top line command if yet another function is wanted.

```
-----------------------  CA-7 CPU JOB DOCUMENTATION  -----------------------
FUNCTION: xxxxxxxx  (APPEND,CLEAR,DELETE,EDIT,FE,FETCH,LIST,REPL,SAVE,UPD)


JOB: xxxxxxxx
SYSTEM:  xxxxxxxx



DESC: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx  LINK: nnnnnn




ACTIVE SIZE: nnnn


PROGRAM: SM11  MSG-INDX: nn  -- DB.4.1   --  yy.ddd / hh:mm:ss
MESSAGE:
```

This panel contains the following fields:

FUNCTION

Specifies the function to perform. Value must be the name of some other panel or one of the following values:

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 panel input data fields and clears the Active Area of the user.

DELETE

Deletes a documentation member from the database.

EDIT

Transfers the user to the edit facility and permits text processing.

FE

Combines FETCH and EDIT.

FETCH

Retrieves documentation data from the database and replaces the Active Area of the user with documentation text.

LIST

Lists documentation data only. In batch, a formatted panel is not listed; only a found or not found message is returned.

REPL

Replaces a documentation member in the database.

SAVE

Adds a documentation member to the database.

UPD

Updates the DESC and the LINK fields only. UPD does not affect the documentation text.

Batch keyword: Positional parameter

n/a

Contains a constant value of JOB for job documentation.

Limits: Required for batch only.

Batch keyword: Positional parameter

JOB

Specifies the job name for which the indicated function is performed.

Limits: 1 to 8 alphanumeric characters

Batch keyword: JOB

SYSTEM

(Optional) Specifies the system name or system ID related to the job for the documentation being defined.

Limits: 1 to 8 alphanumeric characters

Batch keyword: SYSTEM

DESC

(Optional) Specifies an optional description of the job.

Limits: 1 to 45 alphanumeric characters

Batch keyword: DESC

LINK

(Optional) Specifies 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.

Limits: 1 to 6 numeric characters

Batch keyword: LINK

Note: You can create an LPROS loop. Do not link linked documentation to itself. The following example shows a loop:

```
A linked to B
B linked to C
C linked to A
```

ACTIVE SIZE

Identifies a system-generated field that shows the number of lines of text existing in the Active Area for the current terminal session.

More information:

Special Documentation Features (see page 277)
Using the Edit Facility (see page 335)

DB.4.2 - Input/Output Network Documentation Panel

The DB.4.2 - Input/Output Network Documentation panel lets you define network level documentation. Do not use this panel when the SCHD-PROSE field on the DB.5 panel was used for the network. You can define documentation in one continuous stream for the specified network.

Also, you can subdivided documentation into segments and subsegments.

To display the panel, enter:

- 2 as the FUNCTION on the DB.4 Menu panel.
- 4.2 as the FUNCTION on the DB Menu panel.
- DB.4.2 as the FUNCTION on any other menu or formatted input panel.
- DB.4.2 or PROSE,NWK as a top line command.

To exit the panel:

- Press PF3 to return to the DB.4 Menu panel except with the TSO/ISPF interface when PF3 has been assigned to another function.
- Move the cursor to the top line of the panel and enter the wanted top line command if yet another function is wanted.

```
-----------------  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: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx  LINK: nnnnnn

STEP: xxxxxxxx  DDNAME: xxxxxxxx
REPORT-ID: xxxxxxxxxxxxxxxxxxx
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:
```

This panel contains the following fields:

FUNCTION

Specifies the function to perform. Value must be the name of some other panel or one of following values:

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 panel input data fields and clears the Active Area of the user.

DELETE

Deletes a documentation member from the database.

EDIT

Transfers the user to the edit facility and allows text processing.

FE

Combines FETCH and EDIT.

FETCH

Retrieves documentation data from the database and replaces the Active Area of the user with documentation text.

LIST

Lists documentation data only. In batch, a formatted panel is not listed; only a found or not found message is returned.

REPL

Replaces a documentation member in the database.

SAVE

Adds a documentation member to the database.

UPD

Updates formatted panel fields only. UPD does not affect the documentation text.

Batch keyword: Positional parameter

n/a

Contains a constant value of NWK for network documentation.

Limits: Required for batch only.

Batch keyword: Positional parameter

JOB

(Optional) Specifies the job that is related to the network for the documentation being defined.

Limits: 1 to 8 alphanumeric characters

Batch keyword: JOB

SYSTEM

(Optional) Specifies the system name or system ID related to the network for the documentation being defined.

Limits: 1 to 8 alphanumeric characters

Batch keyword: SYSTEM

NETWORK

Specifies the network for which the documentation is being defined.

Limits: 1 to 8 alphanumeric characters

Batch keyword: NETWORK

DESC

(Optional) Specifies a description of the network.

Limits: 1 to 45 alphanumeric characters

Batch keyword: DESC

LINK

(Optional) Specifies 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.

Limits: 1 to 6 numeric characters

Batch keyword: LINK

Note: You can create an LPROS loop. Do not link linked documentation 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:

STEP

(Optional) Specifies the name of a job step in a particular job for which the network documentation is being defined.

Limits: 1 to 8 alphanumeric characters

Batch keyword: STEP

DDNAME

(Optional) Specifies a ddname for which the documentation is being defined.

Limits: 1 to 8 alphanumeric characters

Batch keyword: DD

REPORT-ID

(Optional) Provides identification of a printed report.

Limits: 1 to 20 alphanumeric characters

Batch keyword: REPORT

FORM

(Optional) Specifies the output form number.

Limits: 1 to 8 alphanumeric characters

Batch keyword: FORM

TRAIN

(Optional) Specifies the type of printer train required.

Limits: 1 to 2 alphanumeric characters

Batch keyword: TRAIN

CARRIAGE

(Optional) Specifies the type of carriage control required.

Limits: 1 to 4 alphanumeric characters

Batch keyword: CARR

COPIES

(Optional) Specifies the number of original copies required.

Limits: 1 to 3 numeric characters from 0 to 255

Batch keyword: COPIES

ACTIVE SIZE

Identifies a system-generated field that shows the number of lines of text existing in the Active Area for the current terminal session.

More information:

Special Documentation Features (see page 277)
Using the Edit Facility (see page 335)

DB.4.3 - User-Defined Item Documentation Panel

The DB.4.3 - User-Defined Item Documentation panel lets you define user level documentation.

To display the panel, enter:

- 3 as the FUNCTION on the DB.4 Menu panel.
- 4.3 as the FUNCTION on the DB Menu panel.
- DB.4.3 as the FUNCTION on any other menu or formatted input panel.
- DB.4.3 or PROSE,USER as a top line command.

To exit the panel:

- Press PF3 to return to the DB.4 Menu panel except with the TSO/ISPF interface when PF3 has been assigned to another function.
- Move the cursor to the top line of the panel and enter the wanted top line command if yet another function is wanted.

```
------------------  CA-7 USER-DEFINED ITEM DOCUMENTATION  ------------------
FUNCTION: xxxxxxxx  (APPEND,CLEAR,DELETE,EDIT,FE,FETCH,LIST,REPL,SAVE,UPD)



USER: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx


DESC: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx  LINK: nnnnnn


REPORT-ID: xxxxxxxxxxxxxxxxxxx
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:
```

This panel contains the following fields:

FUNCTION

Specifies the function to perform. Value must be the name of some other panel or one of the following values:

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 panel input data fields and clears the Active Area of the user.

DELETE

Deletes a documentation member from the database.

EDIT

Transfers the user to the EDIT facility and allows text processing.

FE

Combines FETCH and EDIT.

FETCH

Retrieves documentation data from the database and replaces the Active Area of the user with documentation text.

LIST

Lists documentation data only. In batch, a formatted panel is not listed; only a found or not found message is returned.

REPL

Replaces a documentation member in the database.

SAVE

Adds a documentation member to the database.

UPD

Updates formatted panel fields only. It does not affect the documentation text.

Batch keyword: Positional parameter

n/a

Contains a constant value of USER for user documentation.

Limits: Required for batch only.

Batch keyword: Positional parameter

USER

Specifies a required name that is used to supply a unique identifier for the information that the user documentation entry defines. This field must follow z/OS data set naming conventions. The field must not correspond to any system name, job name, data set name, job-step-ddname combination, or workstation name that exists (or will exist) in the CA Workload Automation SE system.

Limits: 1 to 32 alphanumeric characters

Batch keyword: USER

DESC

(Optional) Specifies an optional description of the user entry.

Limits: 1 to 45 alphanumeric characters

Batch keyword: DESC

LINK

(Optional) Specifies 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.

Limits: 1 to 6 numeric characters

Batch keyword: LINK

Note: You can create an LPROS loop. Do not lnk linked documentation 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

(Optional) Provides identification of a printed report.

Limits: 1 to 20 alphanumeric characters

Batch keyword: REPORT

FORM

(Optional) Specifies the output form number.

Limits: 1 to 8 alphanumeric characters

Batch keyword: FORM

TRAIN

(Optional) Specifies the type of printer train required.

Limits: 1 to 2 alphanumeric characters

Batch keyword: TRAIN

CARRIAGE

(Optional) Specifies the type of carriage control required.

Limits: 1 to 4 alphanumeric characters

Batch keyword: CARR

COPIES

(Optional) Specifies the number of original copies required.

Limits: 1 to 3 numeric characters from 0 to 255

Batch keyword: COPIES

ACTIVE SIZE

Identifies a system-generated field that shows the number of lines of text existing in the Active Area for the current terminal session.

More information:

Using the Edit Facility (see page 335)

DB.4.4 - Data Set Documentation Panel

The DB.4.4 - Data Set Documentation panel lets you define data set level documentation. Data sets whose name is longer than 41 characters cannot use this panel.

To display the panel, enter:

- 4 as the FUNCTION on the DB.4 Menu panel.
- 4.4 as the FUNCTION on the DB Menu panel.
- DB.4.4 as the FUNCTION on any other menu or formatted input panel.
- DB.4.4 or PROSE,DSN as a top line command.

To exit the panel:

- Press PF3 to return to the DB.4 Menu panel except with the TSO/ISPF interface when PF3 has been assigned to another function.
- Move the cursor to the top line of the panel and enter the wanted top line command if yet another function is wanted.

```
----------------------  CA-7 DATA SET DOCUMENTATION  -----------------------
FUNCTION: xxxxxxxx  (APPEND,CLEAR,DELETE,EDIT,FE,FETCH,LIST,REPL,SAVE,UPD)




DSN: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx     DSNBR: nnnnnnnn


DESC: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx  LINK: nnnnnn

STEP: xxxxxxxx  DDNAME: xxxxxxxx
REPORT-ID: xxxxxxxxxxxxxxxxxxx
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:
```

This panel contains the following fields:

FUNCTION

Specifies the function to perform. Value must be the name of some other panel or one of the following values:

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 panel input data fields and clears the Active Area of the user.

DELETE

Deletes a documentation member from the database.

EDIT

Transfers the user to the EDIT facility and allows text processing.

FE

Combines FETCH and EDIT.

FETCH

Retrieves documentation data from the database and replaces the Active Area of the user with documentation text.

LIST

Lists documentation data only. In batch, a formatted panel is not listed; only a found or not found message is returned.

REPL

Replaces a documentation member in the database.

SAVE

Adds a documentation member to the database.

UPD

Updates formatted panel fields only. UPD does not affect the documentation text.

Batch keyword: Positional parameter

n/a

Contains a constant value of DSN for data set documentation.

Limits: Required for batch only.

Batch keyword: Positional parameter

DSN

Specifies the data set name for which the documentation is being defined.

Limits: 1 to 41 alphanumeric characters and required unless DSNBR is used.

Batch keyword: DSN

Note: The data set name must not exceed 41 characters to permit PP. to be added as part of the data set name in the IDs.

DSNBR

Specifies the data set for which the documentation is being supplied. Must be the data set number that CA Workload Automation SE assigned when the data set was first defined.

Limits: 1 to 8 numeric characters and required unless DSN is used.

Batch keyword: DSNBR

DESC

(Optional) Specifies an optional description of the data set.

Limits: 1 to 45 alphanumeric characters

Batch keyword: DESC

LINK

(Optional) Specifies 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.

Limits: 1 to 6 numeric characters

Batch keyword: LINK

Note: You can create an LPROS loop. Do not link linked documentation 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:

STEP

(Optional) Specifies the name of a job step in a particular job for which the documentation is being defined.

Limits: 1 to 8 alphanumeric characters

Batch keyword: STEP

DDNAME

(Optional) Specifies a ddname for which the documentation is being defined.

Limits: 1 to 8 alphanumeric characters

Batch keyword: DD

REPORT-ID

(Optional) Provides identification of a printed report.

Limits: 1 to 20 alphanumeric characters

Batch keyword: REPORT

FORM

(Optional) Specifies the output form number.

Limits: 1 to 8 alphanumeric characters

Batch keyword: FORM

TRAIN

(Optional) Specifies the type of printer train required.

Limits: 1 to 2 alphanumeric characters

Batch keyword: TRAIN

CARRIAGE

(Optional) Specifies the type of carriage control required.

Limits: 1 to 4 alphanumeric characters

Batch keyword: CARR

COPIES

(Optional) Specifies the number of original copies required.

Limits: 1 to 3 numeric characters from 0 to 255

Batch keyword: COPIES

ACTIVE SIZE

Identifies a system-generated field that shows the number of lines of text existing in the Active Area for the current terminal session.

More information:

Using the Edit Facility (see page 335)

DB.4.5 - DD Statement Documentation Panel

The DB.4.5 - DD Statement Documentation panel lets you define DD level documentation.

To display the panel, enter:

- 5 as the FUNCTION on the DB.4 Menu panel.
- 4.5 as the FUNCTION on the DB Menu panel.
- DB.4.5 as the FUNCTION on any other menu or formatted input panel.
- DB.4.5 or PROSE,DD as a top line command.

To exit the panel:

- Press PF3 to return to the DB.4 Menu panel except with the TSO/ISPF interface when PF3 has been assigned to another function.
- Move the cursor to the top line of the panel and enter the wanted top line command if yet another function is wanted.

```
---------------------  CA-7 DD STATEMENT DOCUMENTATION  ----------------------
FUNCTION: xxxxxxxx  (APPEND,CLEAR,DELETE,EDIT,FE,FETCH,LIST,REPL,SAVE,UPD)


JOB: xxxxxxxx
SYSTEM: xxxxxxxx




DESC: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx  LINK: nnnnnn

STEP: xxxxxxxx  DDNAME: xxxxxxxx
REPORT-ID: xxxxxxxxxxxxxxxxxxx
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:
```

This panel contains the following fields:

FUNCTION

Specifies the function to perform. Value must be the name of some other panel or one of the following values:

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 panel input data fields and clears the Active Area of the user.

DELETE

Deletes a documentation member from the database.

EDIT

Transfers the user to the EDIT facility and allows text processing.

FE

Combines FETCH and EDIT.

FETCH

Retrieves documentation data from the database and replaces the Active Area of the user with documentation text.

LIST

Lists documentation data only. In batch, a formatted panel is not listed; only a found or not found message is returned.

REPL

Replaces a documentation member in the database.

SAVE

Adds a documentation member to the database.

UPD

Updates formatted panel fields only. UPD does not affect the documentation text.

Batch keyword: Positional parameter

n/a

Contains a constant value of DD for the DD documentation.

Limits: Required for batch only.

Batch keyword: Positional parameter

JOB

Specifies the job containing the ddname for the documentation being defined.

Limits: 1 to 8 alphanumeric characters

Batch keyword: JOB

SYSTEM

(Optional) Specifies the system name or system ID for which the documentation is being defined.

Limits: 1 to 8 alphanumeric characters

Batch keyword: SYSTEM

DESC

(Optional) Specifies a description of the data set.

Limits: 1 to 45 alphanumeric characters

Batch keyword: DESC

LINK

(Optional) Specifies 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.

Limits: 1 to 6 numeric characters

Batch keyword: LINK

Note: You can create an LPROS loop. Do not link linked documentation to itself. The following example shows a loop:

```
A linked to B
B linked to C
C linked to A
```

STEP

Specifies the step of the JOB for the DD documentation being defined.

Limits: 1 to 8 alphanumeric characters

Batch keyword: STEP

DDNAME

Specifies the ddname for which the documentation is being defined.

Limits: 1 to 8 alphanumeric characters

Batch keyword: DD

The following fields are informational only:

REPORT-ID

(Optional) Provides identification of a printed report.

Limits: 1 to 20 alphanumeric characters

Batch keyword: REPORT

FORM

(Optional) Specifies the output form number.

Limits: 1 to 8 alphanumeric characters

Batch keyword: FORM

TRAIN

(Optional) Specifies the type of printer train required.

Limits: 1 to 2 alphanumeric characters

Batch keyword: TRAIN

CARRIAGE

(Optional) Specifies the type of carriage control required.

Limits: 1 to 4 alphanumeric characters

Batch keyword: CARR

COPIES

(Optional) Specifies the number of original copies required.

Limits: 1 to 3 numeric characters from 0 to 255

Batch keyword: COPIES

ACTIVE SIZE

Identifies a system-generated field that shows the number of lines of text existing in the Active Area for the current terminal session.

More information:

Using the Edit Facility (see page 335)

DB.4.6 - Application System Documentation Panel

The DB.4.6 - Application System Documentation panel lets you define system level documentation.

To display the panel, enter:

- 6 as the FUNCTION on the DB.4 Menu panel.
- 4.6 as the FUNCTION on the DB Menu panel.
- DB.4.6 as the FUNCTION on any other menu or formatted input panel.
- DB.4.6 or PROSE,SYS as a top line command.

To exit the panel:

- Press PF3 to return to the DB.4 Menu panel except with the TSO/ISPF interface when PF3 has been assigned to another function.
- Move the cursor to the top line of the panel and enter the wanted top line command if yet another function is wanted.

```
------------------  CA-7 APPLICATION SYSTEM DOCUMENTATION  -------------------
FUNCTION: xxxxxxxx  (APPEND,CLEAR,DELETE,EDIT,FE,FETCH,LIST,REPL,SAVE,UPD)


SYSTEM: xxxxxxxx



DESC: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx  LINK: nnnnnn




ACTIVE SIZE: nnnn

PROGRAM: SM11  MSG-INDX: nn  -- DB.4.6   --  yy.ddd / hh:mm:ss
MESSAGE:
```

This panel contains the following fields:

FUNCTION

Specifies the function to perform. Value must be the name of some other panel or one of the following values:

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 panel input data fields and clears the Active Area of the user.

DELETE

Deletes a documentation member from the database.

EDIT

Transfers the user to the EDIT facility and allows text processing.

FE

Combines FETCH and EDIT.

FETCH

Retrieves documentation data from the database and replaces the Active Area of the user with documentation text.

LIST

Lists documentation data only. In batch, a formatted panel is not listed; only a found or not found message is returned.

REPL

Replaces a documentation member in the database.

SAVE

Adds a documentation member to the database.

UPD

Updates the DESC and the LINK fields only. UPD does not affect the documentation text.

Batch keyword: Positional parameter

n/a

Contains a constant value of SYS for system documentation.

Limits: Required for batch only.

Batch keyword: Positional parameter

SYSTEM

Specifies the system name or system ID for which the documentation entry is being defined.

Limits: 1 to 8 alphanumeric characters

Batch keyword: SYSTEM

DESC

(Optional) Specifies an optional description of the system.

Limits: 1 to 45 alphanumeric characters

Batch keyword: DESC

LINK

(Optional) Specifies 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.

Limits: 1 to 6 numeric characters

Batch keyword: LINK

Note: You can create an LPROS loop. Do not link linked documentation to itself. The following example shows a loop:

```
A linked to B
B linked to C
C linked to A
```

ACTIVE SIZE

Identifies a system-generated field that shows the number of lines of text existing in the Active Area for the current terminal session.

More information:

Using the Edit Facility (see page 335)

Special Documentation Features

CA Workload Automation SE has six major categories of user-defined documentation. Each category has its own DB.4 panel with which the documentation can 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

User-Defined Documentation Segments

Within each of the major documentation member types, the user can subdivide documentation text into segments. Segments are convenient because you can access them 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.

Reserved Segment Names

The system reserves some names for special use. Do not use segment names that would conflict with the following segment names:

#END

#QDESC

#RESTART

#station-name

Discussions of these special purpose names follow later.

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 method adds ease of retrieval.
- 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 eight characters.
  - Commas, as part of the name, are not allowed.
  - Alphabetic, numeric, and special characters are permitted.
- A segment name must be unique within the documentation member. Make the segment name different from the documentation member name of which the segment is a part.

More information:

Naming Conventions Documentation (see page 284)

Segment Names Definition

The user-defined segments can be an orderly method of documentation. By establishing standards that are related to segment names, a powerful tool results. To define a segment name, select an eight-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 list contains potential segment names that could fit into the naming conventions for segments:

#DATECARD

Defines the format of the date cards that are used in this job.

#progname

Provides a description of programs that are used in the job.

#stepname

Provides a description of job steps in the job.

#DISASTER

Defines the instructions for disaster recovery procedures.

#RECOVERY

Defines the recovery specifications for I/O errors, abends, and so forth.

#FILING

Provides the instructions for filing input or output.

#SETUP

Defines I/O media requirements of the job.

#CONTACTS

Defines names and numbers of personnel who support the job.

#DEVICES

Defines I/O device requirements of the job.

#CONFLICT

Defines possible processing conflicts with other jobs.

#VALIDATE

Defines special manual verification requirements of the job.

#AFTERHRS

Defines special procedures occurring after the close of the business day.

Other segment names can be helpful or more meaningful and easy to use in any particular data center. Within the constraints of the few reserved segment names, users have total flexibility in naming documentation segments.

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 that is 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. The system 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, you must have a #END control statement. The #END statement 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
```

Documentation Segments Display

If your installation has adopted standard naming conventions, displaying the documentation that is 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 more documentation on this job is available, it is not displayed. Only the requested segment 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.

Subsegments

Each segment can contain subsegments. The rules and guidelines that apply to subsegments are similar to the ones for segments.

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.

Each subsegment must have a name. Subsegment names must follow the same rules as segment names.

A segment can 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 must not have the same name as a reserved segment name.

A subsegment must be embedded between a #subsegment-name statement and a #END,subsegment-name statement.

More information:

Reserved Segment Names (see page 278)
Rules and Guidelines for Documentation Segmentation (see page 278)
Naming Conventions Documentation (see page 284)

Subsegment Names Definition

To define a subsegment name, select an eight-character name meaningful in your environment; precede the name with a # sign. Enter the definition starting in column 1. This method is the same way segment names are defined:

```
#subsegment-name
```

For example, your installation can have two procedures for recovery: one for business hours and one for nonbusiness hours. To define these two names, you can create two subsegments within the segment that is named RECOVERY using the following subsegment names:

```
#DAYTIME
#AFTERHRS
```

Format of Documentation Subsegments

The format of documentation subsegments is exactly like the format of segments. Make the first statement of the subsegment #subsegment-name, and begin in column 1. Immediately following this statement is the subsegment documentation that 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. The system then 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, you must have a #END,subsegment-name control statement. Nesting of subsegments within other subsegments is not permitted. 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
```

Documentation Subsegments Display

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 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
```

Naming Conventions Documentation

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
```

The DB.4.3 panel contains more information about the USER field.

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 members can be maintained easily through the appropriate DB.4 panels.

More information:

DB.4.3 - User-Defined Item Documentation Panel (see page 262)

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 can 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 panel (and the EDIT facility).

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
```

#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 panel (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 can be listed by the LPROS transaction with job name or network specified.

Note: Module SASSMSGS can cause suppression of the #QDESC. Check messages SFEP-11 and SIRD-11 in the SASSMSGS module.

#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 wanted station names are defined as part of the documentation with a #station-name statement. The station name that 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 can be entered between points (A) and (B), but these lines would only be displayed through the LPROS command or DB.4 panels. 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 SASSMSGS 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.

#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
```

#END Statement

This statement is used to signal the end of a documentation unit.

This statement has the following format:

```
  #END[,segment-name]
       [,subsegment-name]
       [,station-name]
```

#END

Defines the required statement name. 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

Defines 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 can be nested within one segment of documentation.

More information:

User-Defined Documentation Segments (see page 278)

Chapter 8: Workstation Networks

This section describes setting up and maintaining a workstation network within CA Workload Automation SE.

This section contains the following topics:

Network Adds (see page 291)
DB.5 - Input/Output Network Definition Panel (see page 292)
Network Changes (see page 295)

Network Adds

Adding a network is similar to adding a job. The primary method is to use the ADD function on the DB.5 panel. 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, writing a program to produce batch terminal network commands can be an advantage. The database can then be updated using the Batch Terminal Interface (BTI) facility.

Note: For more information about BTI, the *Interface Reference Guide*.

The following items are something to remember when adding a network:

- On the DB.5 panel, stations must be defined in the chronological order in which they are used.
- The DMDNW and RUNNW commands do not add a network to the database. These commands require that the network is defined in the database with some type of schedule before the commands can act on the network.
- 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 panel functions.
- Input networks that are not handled on an on-request basis must have a calendar-based schedule defined. This definition is done with the DB.2.2 panel.
- 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.
- Output networks must be logically connected with the DB.3.4 panel. Scheduling parameters must be entered through the DB.2.3 panel, but the schedule is not resolved against a calendar.

(More information:

Batch Input (see page 35)

DB.5 - Input/Output Network Definition Panel

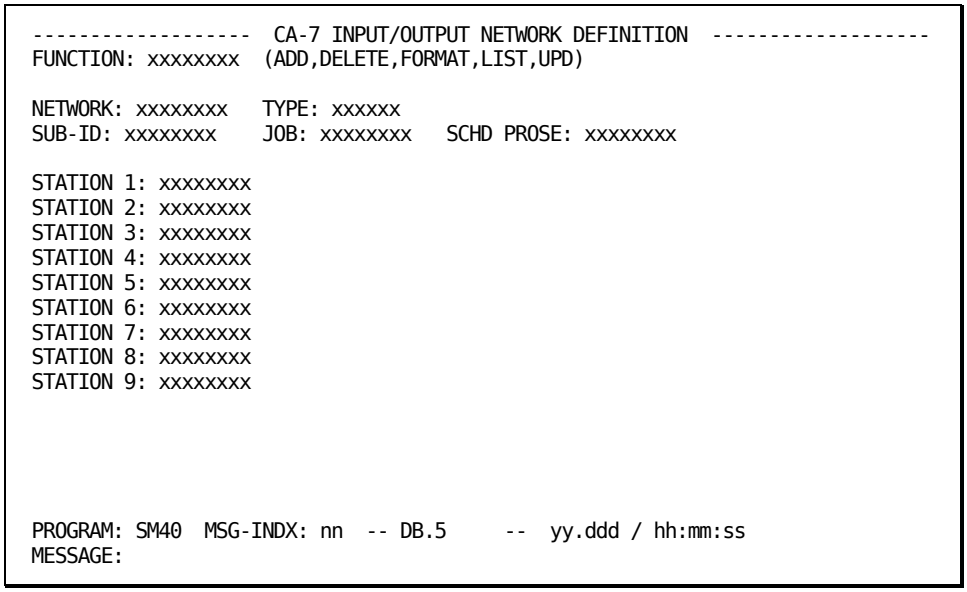
The Input/Output Network Definition panel lets you define networks of workstations at which manual tasks are performed.

To display the panel, enter:

- 5 as the FUNCTION on the DB Menu panel.
- DB.5 as the FUNCTION value on any other menu or formatted input panel.
- DB.5 or NETWORK as a top line command.

To exit the panel:

- Press PF3 to return to the DB Menu panel except with the TSO-ISPF interface when PF3 has been assigned to another function.
- Move the cursor to the top line of the panel and enter the wanted top line command if yet another function is wanted.



This panel contains the following fields:

FUNCTION

Specifies the function to perform. Value must be the name of some other panel 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 when it is connected to a job.

FORMAT

Clears the panel of all user input data.

LIST

Lists the database information. In batch, a formatted panel is not listed; only a found or not found message is returned.

UPD

Updates network information in the database.

Batch keyword: Positional parameter

NETWORK

Specifies the network on which to perform the indicated function. Must be a network name.

Limits: 1 to 8 alphanumeric characters

Batch keyword: Positional parameter

TYPE

Specifies the value must be INPUT or OUTPUT to indicate the type of network.

Limits: Required for ADD and UPD functions. Optional for DELETE, FORMAT, and LIST functions.

Batch keyword: Positional parameter

SUB-ID

(Optional) Specifies further description of the network's duties.

Limits: 1 to 8 alphanumeric characters

Batch keyword: SUBID

Output Networks: Specify the SUBID 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 task number of the user, job number, or control number for the data can be defined as the sub-ID.

JOB

(Optional) Specifies 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.

Limits: 1 to 8 alphanumeric characters

Batch keyword: JOB

SCHD PROSE

(Optional) Designates the name of some other network whose DB.4.2 member, #QDESC segment, is used for special documentation messages. Schedule Scan prints the messages when this network enters the queue. Cannot be used if this network has a DB.4.2 member that is defined for it.

Limits: 1 to 8 alphanumeric characters

Batch keyword: SCHPROSE

STATION

Enables entry of up to nine workstations into this network. Enter the station names in the sequence in which the corresponding workstation tasks are performed. The same station name can 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, make the station names a sublist. For example:

STAT=(st1,st2,st3)

Limits: 1 to 8 alphanumeric characters, and at least one entry is required.

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, an update cannot change the number of stations in a network.

Example: Batch Input

This batch input example adds input network PAYPUNCH to the CA Workload Automation SE database. The network consists of three stations: LOG, KEY, and VERIFY.

```
NETWORK
ADD,PAYPUNCH,INPUT,STAT=(LOG,KEY,VERIFY)
DBM
```

Network Changes

If the characteristics of the network are changing from those shown on the DB.5 panel, use the UPD function on the DB.5 panel. If you have many changes, use the Batch Terminal Interface (BTI) facility to make these changes.

Note: For more information about BTI, see the *Interface Reference 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 panels 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 panel 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.

More information:

Batch Input (see page 35)

Chapter 9: Data Sets

This section contains instructions for adding and maintaining data set information in CA Workload Automation SE.

This section contains the following topics:

Data Set Adds (see page 297)
DB.6 - Data Set Definition Panel (see page 299)
Data Set Changes (see page 304)

Data Set Adds

Several methods of adding data sets to the database are available. 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 LOADDSNS keyword is used on the DBASE statement in the initialization file, the LOAD process does not build any DD or data set information for jobs that are marked MAINT=Y on the DB.1 (JOB) panel.

Data sets can also be added to the database by using the DB.6 panel ADD function.

If data set information exists in machine-readable form, a program can be written to perform the DB.6 panel functions with batch commands and the Batch Terminal Interface (BTI) facility. For further details on BTI, see the *Interface Reference Guide*.

Once a data set is added to the database, CA Workload Automation SE assigns a unique eight-digit number to the data set, referred to as a DSNBR. You can then reference the data set by that number. This method eliminates the need to enter a long data set name each time that CA Workload Automation SE commands and panels reference a data set.

More information:

Batch Input (see page 35)

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 CA Workload Automation SE can submit the job. Some data sets, however, are not created by a CA Workload Automation SE submitted job. The user may want these data sets to be considered as permanent to CA Workload Automation SE.

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

  The reserved ddnames reside in the SASSPMDD table. Names can be added or deleted as necessary to satisfy the needs of the user. For more information about modifying this table, see the topic Reserved DDNAME Table in the chapter "User Exits and Modifications" of the *Systems Programming Guide*.

The TYPE field on the DB.6 panel must be set to PERM for permanent data sets to CA Workload Automation SE. This action 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.

Dynamically Allocated Data Sets

Data sets that are dynamically allocated by the programs and do not have a DD statement in the JCL must be defined with the DB.6 panel. The LOAD process cannot perform this function since a DD statement was not used.

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 the control of CA Workload Automation SE. Such data sets are referred to as external data sets and must be defined using the DB.6 panel. Data sets that are used by jobs under the control of CA Workload Automation SE 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 panel, but the requirements must be satisfied by the user before the dependent job can execute. Requirement posting can be performed using online panels, batch terminal interface commands, Trailer Step commands, or U7SVC input.

DB.6 - Data Set Definition Panel

The DB.6 - Data Set Definition panel lets you define a user data set to CA Workload Automation SE.

Normally, you use this panel to define data sets with a TYPE value of PERM or for updates. If this panel 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 that is required on this information with this panel.

To display the panel, enter:

- 6 as the FUNCTION on the DB Menu panel.
- DB.6 as the FUNCTION value on any other menu or formatted input panel.
- DB.6 or DSN as a top line command.

To exit the panel:

- Press PF3 to return to the DB Menu panel except with the TSO-ISPF interface when PF3 has been assigned to another function.
- Move the cursor to the top line of the panel and enter the wanted top line command if yet another function is wanted.

```
------------------------  CA-7 DATA SET DEFINITION  ------------------------
FUNCTION: xxxxxxxx  (ADD,DELETE,FORMAT,LIST,RENAME,UPD)

DSN: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx  DSNBR: nnnnnnnn

NEWNAME: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

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:
```

This panel contains the following fields:

FUNCTION

Specifies the function to perform. Value must be the name of some other panel 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 panel of all user input data.

LIST

Lists the formatted data of the panel. In batch, a formatted panel 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 can be used to change the CA Workload Automation SE data set name. The RENAME top line utility command changes the data set name in z/OS; whereas, the RENAME function on this panel changes the data set name in the CA Workload Automation SE database, but not in z/OS.

UPD

Updates data set information in the database.

Batch keyword: Positional parameter

DSN

(Optional) Specifies the data set and must be the fully qualified data set name.

Limits: 1 to 44 alphanumeric characters and required for ADD (and UPD unless DSNBR is used).

Batch keyword: DSN

Note: DSNs with embedded blanks in the name or any other nonconventional data set names are not supported.

DSNBR

(Optional) Specifies the data set to use. The value must be the numeric value that CA Workload Automation SE already assigned.

Limits: 1 to 8 numeric characters (leading zeros are not required) and required unless DSN is used (ignored for ADD)

Batch keyword: DSNBR

NEWNAME

(Optional) Used with the RENAME function and DSN field to change a data set name in the CA Workload Automation SE database. The RENAME top line utility command changes the data set name in z/OS; whereas, the RENAME function on this panel changes the data set name in the CA Workload Automation SE database, but not in z/OS.

Limits: 1 to 44 alphanumeric characters

Batch keyword: NEWNAME

TYPE

(Optional) Specifies the data set type. Value can be one of the following times:

NORM

Internal: Specifies both the creating-job and using-jobs for this data set are known to CA Workload Automation SE.

External: Specifies the creating-job, using-jobs, or both for this data set are not known to CA Workload Automation SE.

PERM

Specifies this data set is always available for input.

Important! When a data set is marked PERM, no SMF records are captured. PERM means no data set triggering can occur with this type data set.

Default: NORM when doing ADD

Batch keyword: TYPE

GDG

(Optional) Specifies whether this data set is a generation data group (Y or N).

Default: N

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

(Optional) Specifies whether the interface to System Management Facility (SMF) inserts the values for DSORG, RECFM, LRECL, and BLKSIZE when this data set is next used (Y or N). Unless the value is reset, this insertion is only done once.

Default: Y

Batch keyword: SMF

Note: After the first time the job that creates this data set runs under CA Workload Automation SE, this field is reset to N. If 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 process causes CA Workload Automation SE to record the new values when the job is run again.

POST AT CLOSE TIME

(Optional) Specifies 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 process does not ensure successful step or job termination.

Limits: Y or N

Default: N (causes triggering or posting action to take place at the job termination)

Batch keyword: POST

DEVICE

(Optional) Specifies a device type. If specified, TAPE or DASD are the only acceptable values.

Batch keyword: DEVICE

DSORG

(Optional) Specifies the data set organization. Same as the DSORG specified in the DCB subparameter in the JCL.

Limits: 1 to 3 alphanumeric characters

Batch keyword: DSORG

RECFM

(Optional) Specifies the record format of the data set. Same as the RECFM specified in the DCB subparameter in the JCL.

Limits: 1 to 5 alphanumeric characters

Batch keyword: RECFM

LRECL

(Optional) Specifies the logical record length of the data set. Same as the LRECL specified in the DCB subparameter in the JCL.

Limits: 1 to 5 numeric characters

Batch keyword: LRECL

BLKSIZE

(Optional) Specifies the block size of the data set. Same as the BLKSIZE specified in the DCB subparameter in the JCL.

Limits: 1 to 5 numeric characters

Batch keyword: BLKSIZE

More information:

Permanent Data Sets (see page 298)

Usage Notes

You can maintain connections to jobs that require this data set using the DB.3.1 panel.

Example: Batch Input

This batch input example adds data set SYS2.PROCLIB to the CA Workload Automation SE database as a permanent type.

```
DBM
DSN
ADD,DSN=SYS2.PROCLIB,TYPE=PERM
DBM
```

Data Set Changes

Use the DB.6 panel 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 panel is that of renaming data sets. If the DCB for this data set is not changing, all that you must enter is the RENAME function and the NEWNAME for the data set. This panel changes the data set name in the CA Workload Automation SE database. To change the data set name in z/OS, use the RENAME top line utility command.

If the DCB attributes change, be sure that the SMF indicator has the proper value for the situation. If the system is to pick up automatically the DCB characteristics, set the SMF indicator to Y. The appropriate DCB field cleared to blanks or zeros.

Note: Ensure that the RENAME occurs before changing JCL and loading of jobs. If not, the DB.3.1 updates sometimes need to be manually performed to disconnect the old data set name.

Chapter 10: JCL and PARM Data Maintenance

This section identifies the panels and commands used to manage JCL and PARM data within CA Workload Automation SE.

This section contains the following topics:

Parameter Data Maintenance for Internal Cross-Platform Job Types (see page 305)
DB.7 - JCL Library Maintenance Panel (see page 306)
Scheduled Overrides (see page 311)
Additional Override Statements (see page 318)
SASSJCLU - JCL Utility (see page 328)
JCL Validation (see page 330)
LOAD Command Processing (see page 331)
Special Override Library (see page 331)
Alternate JCL Libraries (see page 333)

Parameter Data Maintenance for Internal Cross-Platform Job Types

Although these topics consistently refer to JCL (for CPU job types), the topics also apply to parameter data (PARM data) for XPJOB and agent job types. XPJOB and agent jobs are referred to as internal cross-platform jobs. Unless a specific note for XPJOBs or agent jobs is present, you can apply all JCL concepts that are discussed here to internal cross-platform jobs. You should also be aware of several other items.

When using the JCL or QJCL commands or panels to edit CA7TOUNI or internal cross-platform PARM data, the FETCH AND EDIT (FE) function is best used to maintain and allow case-sensitive editing on the parameter data. With the FE function, CA Workload Automation SE accesses the database or queue record that denotes whether the job is a CA7TOUNI or internal cross-platform job type. This function enforces the mixed-case editing of the data. If the data is fetched using the FETCH function and then edited using the EDIT function, the information about the job type is lost. The editor defaults to the OPTIONS,INITCASE specification in the ONLINE initialization file.

Unless an internal cross-platform job has been previously defined in the CA Workload Automation SE database, the initialization file OPTIONS statement, INITCASE keyword, determines the case-sensitivity for the edit session. If the internal cross-platform job has been previously defined, the mixed case attribute (which allows both uppercase and lowercase letters) is automatically assigned. If INITCASE=N is coded, all data is forced to uppercase. If INITCASE=Y is coded, you can set the case by entering MIXED or UPPER in the command area of the edit session.

If a PARM data member is added through this maintenance facility, this information is not added to the internal cross-platform job definition. If you want to use the new member during the submission process, you must update the internal cross-platform job definition through the appropriate formatted screen using a top line command (XPJOB or AGJOB).

Any references to the LOAD process do not apply to internal cross-platform job types. Because these jobs only have one step and do not reference any z/OS data sets, the LOAD process is not executed for internal cross-platform jobs.

The RUN and RUNH functions do not work for internal cross-platform job types from the JCL Maintenance panels because required information (such as Node, Agent Job Type, Agent Name, and Executable (file or script)) is not contained within the PARM data. Because these functions can be used for JCL of the CPU jobs that are not defined in the CA Workload Automation SE database, the functions can be attempted, and JCL errors result for internal cross-platform jobs.

Regular jobs can have either a JCL-ID or a JCLLIB coded. JCL-ID is numeric, and JCLLIB is symbolic. The JCL Maintenance panel contains both of these fields. Internal cross-platform jobs use one field for both numeric and symbolic values. This field is named PARMLIB (XPJOBs include the word "Optional"). Its batch keyword equivalent is PRMLIB. If the internal cross-platform job uses a numeric value for PRMLIB, put that value in the JCL Maintenance panel JCL-ID field. If the PRMLIB value uses a symbolic name (starting with &), use that symbolic name in the JCL Maintenance panel JCLLIB field.

DB.7 - JCL Library Maintenance Panel

The DB.7 - Library Maintenance panel lets you enter or review statements in PDS or sequential libraries. It can also be used to review CA Librarian or CA Panvalet for z/OS JCL.

To display the panel, enter:

- 7 as the FUNCTION on the DB Menu panel.
- DB.7 as the FUNCTION on any other menu or formatted input panel.
- DB.7 or JCL as a top line command.

To exit the panel:

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

```
----------------------  CA-7 JCL LIBRARY MAINTENANCE  -----------------------
FUNCTION:  xxxxxxxx  (APPEND,CLEAR,DELETE,EDIT,FE,FETCH,
                     RENAME,REPL,RUN,RUNH,SAVE)

MEMBER:  xxxxxxxx   NEWNAME/OPTION: xxxxxxxx

DSN:  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx  JCL-ID: nnn
VOLSER:  xxxxxx                                        JCLLIB: &xxxxxxxxxxxxxxx

ACTIVE SIZE:  nnnn









PROGRAM:  SM50  MSG-INDX: nn  -- DB.7     --  yy.ddd / hh:mm:ss
MESSAGE:
```

This panel contains the following fields:

FUNCTION

Specifies the function to perform. Value must be the name of some other panel 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 panel 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. If editing a CA7TOUNI or internal cross-platform job, we recommend the FE function to ensure a mixed-case editing environment.

FE

Combines FETCH and EDIT commands.

FETCH

Retrieves data and replaces the user's Active Area with this text. If editing a CA7TOUNI or internal cross-platform job, we recommend the FE function to ensure a mixed-case editing environment.

RENAME

Used with the NEWNAME field, renames 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 that 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.

Limits: RUN(H) function is not valid with internal cross-platform job types.

Note: The SASSXX05 exit is invoked for each JCL statement. Also, if a job is in the CA Workload Automation SE database with this member name and it is set to insert the CA WA Restart Option RMS step and the CA WA Restart Option interface is installed, the RMS step is inserted. However, no other job definition (DB.1) values are used for the current submission.

RUNH

Performs the same action as RUN function except this job is placed in CA Workload Automation SE 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 for z/OS files.

Batch keyword: Positional parameter

MEMBER

Specifies the required member name for PDS, CA Panvalet for z/OS, or CA Librarian access. This field must be omitted for sequential files.

Limits: 1 to 8 alphanumeric characters

Batch keyword: Positional parameter

NEWNAME/OPTION

(Optional) Specifies a multiple purpose field. This field is required with the RENAME function to define the new name that is to replace the old name.

Default: Blank

Limits: 1 to 8 alphanumeric characters and required for RENAME function only.

Batch keyword: NEWNAME

Note: For CA Librarian or CA Panvalet for z/OS data set functions, a value of N indicates that *include* statements are *not* to be expanded. Default causes those statements to be expanded.

DSN

(Optional) Specifies the optional data set name. When reviewing a member of a PDS, CA Panvalet for z/OS, 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 that follows.

Limits: 1 to 44 alphanumeric characters

Batch keyword: DSN

JCL-ID

(Optional) Specifies the 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.

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).

Limits: 1 to 3 numeric characters from 0 to 254. A value of 255 is reserved for symbolic indexes. See JCLLIB.

Batch keyword: JCLID

VOLSER

(Optional) Specifies the volume serial number on which the data set resides. Only used with DSN (that precedes). The volume must be available to CA Workload Automation SE.

Limits: 1 to 6 alphanumeric characters

Batch keyword: VOL

JCLLIB

(Optional) Specifies the 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.

Default: Blank

Limits: 2 to 16 alphanumeric characters beginning with an ampersand (&). The value &HELP is reserved.

Batch keyword: JCLLIB

ACTIVE SIZE

Indicates a system-generated field that tells the user how many lines exist in the Active Area for the current terminal session.

Note: CA Panvalet for z/OS, CA Librarian, and PDS data sets require a JCL definition in the initialization file and in the execution JCL for CA Workload Automation SE.

When EDIT or FE is entered, the user is transferred to the edit facility.

More information:

CA 7 Text Editor Environment (see page 338)

Usage Notes

A shortcut method of displaying a specific job is to enter the following as a top line command:

JCL,*jobname*

You can use this panel to access any card-image PDS, CA Librarian, [assign the PANVLT for your book], or sequential file that is on a DASD volume available to CA Workload Automation SE.

If a SAVE/REPL is done for a member name that matches a job name in the CA Workload Automation SE database, and the specified DSN library is a CA Workload Automation SE JCL data set, the job is flagged for reload on the next run. However, if RELOAD of N is used on the DB.1 (JOB) panel, the RELOAD on the next run is ignored.

When editing CA7TOUNI or internal cross-platform jobs, we recommend the FE (fetch and edit) function to preserve the mixed case editing environment.

Note for the RUN Function: If external security is being used with SUBUID, a job that is not defined to CA Workload Automation SE (not in the CA Workload Automation SE database) has the owner user ID to be considered the same as the requestor user ID.

Scheduled Overrides

CA Workload Automation SE lets you schedule JCL overrides. The scheduled overrides can consist of any statement that can be included with the JCL for a job, not only JCL statements. Some special purpose statements, described later in this section, and any JCL or embedded data records can be scheduled. This method enables users to have the run stream contents dynamically constructed to fit the needs of any particular run.

You can schedule overrides for use:

- 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

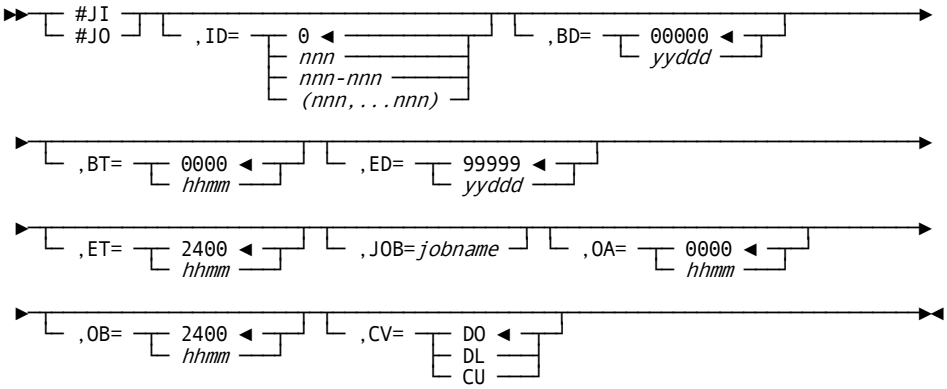
This method lets users prepare the proper statements anytime in advance of their needs without having to remember them on the days they are needed. This method also enables users to ensure that they are used only on the appropriate days.

The scheduled overrides are placed in the execution JCL members. They cannot be used in the PROCLIB members because CA Workload Automation SE 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 Workload Automation SE control statements. One special statement is placed immediately ahead of the statements to define the scheduling criteria. Another special statement is placed at the end of the *set* to mark the end of the statements being scheduled. Multiple *sets* can 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 to include or exclude the other statements in the set based on those criteria. The characters #JI, #JO, #XI, or #XO in positions 1-3 of the statement identify these special statements. Also, these statements cannot be continued.

#JI and #JO Statements

These statements have the following format:



#JI

Indicates to *I*nclude the statements based on the specified scheduling criteria.

#JO

Indicates to *O*mit the statements based on the specified scheduling criteria.

ID

(Optional) Specifies a schedule ID number. If the job is scheduled by this ID, this test for statement inclusion or exclusion is considered true.

Default: 0

Limits: 1 to 3 decimal characters from 0 to 255

0

All schedule IDs.

nnn

Defines a specific schedule ID.

nnn-nnn

Defines 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*)

Defines 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

(Optional) Specifies a *b*eginning *d*ate after which the overrides are in effect.

Default: 00000

Limits: 5 numeric characters Julian date specified as *yyddd*

BT

(Optional) Specifies a *b*eginning *t*ime-of-day for the BD date.

Default: 0000

Limits: 4 numeric characters specified as *hhmm*

ED

(Optional) Specifies an *e*nding *d*ate after which the overrides are no longer in effect.

Default: 99999

Limits: 5 numeric characters Julian date specified as *yyddd*

ET

(Optional) Specifies an *e*nding *t*ime-of-day for the ED date.

Default: 2400

Limits: 4 numeric characters specified as *hhmm*

JOB

(Optional) Specifies that inclusion/exclusion is to be based on a matching fully qualified job name.

Limits: 1 to 8 alphanumeric characters

OA

(Optional) Specifies a time-of-day *o*nly at which and *a*fter which the overrides are to be considered for inclusion or exclusion. The OA time should be lower than the OB time specified.

Default: 0000

Limits: 4 numeric characters specified as *hhmm*

OB

(Optional) Specifies a time-of-day *o*nly at which and *b*efore which the overrides are to be considered for inclusion or exclusion.

Default: 2400

Limits: 4 numeric characters specified as *hhmm*

CV

(Optional) Specifies the *c*omparison *v*alues against which BD, BT, ED, ET, OA, and OB parameters are to be compared.

Default: DO

DO

Specifies to compare values against the *d*ue *o*ut date and the time-of-day values.

DL

Specifies to compare values against the *d*ead*l*ine date and the time-of-day values.

CU

Specifies to compare values against the *cu*rrent date and the time-of-day values.

The ID, BD, BT, ED, ET, OA, OB, and CV parameters can be used in combination to accomplish the wanted 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, LEADTM, or both 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.

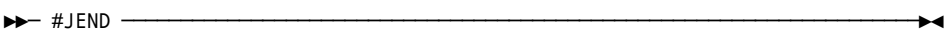
If any errors are encountered in the #JI or #JO override statements (that is, invalid data for ID, BD, BT, ED, ET, OA, OB, or CV fields), a message is sent notifying the MASTER station that the JCL could not be attached, and the job remains in the request queue in SKELETON status.

The first statement in any JCL member must be a JOB statement and cannot be overridden. The exception is a /*PRIORITY statement that can precede the JOB statement. However, #JI and #JO statements must come after the JOB statement.

#JEND Statement

Another #JO or #JI statement, an end-of-file, or a #JEND statement can terminate the conditional statements. The #JEND statement indicates the end of a set of Scheduled Override control statements.

This statement has the following format:



#JEND has 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.

#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 before JCL submission, #X statements can be reviewed, changed, or both with the QM.5 panel.

Scheduled Override Examples

Example: Omit statements with specific schedule ID

This example omits statements whenever a job is scheduled by Schedule ID 6.

```
#JO,ID=6
  .
  . (JCL statements to be omitted)
  .
#JEND
```

Example: Include statements with date and time

This example includes statements until date 13265 at 8.00 a.m.

```
#JI,ED=13265,ET=0800
  .
  . (JCL statements to be included)
  .
#JEND
```

Example: Omit statements with dates

This example omits statements from 13200 through 13206.

```
#JO,BD=13200,ED=13206
  .
  . (JCL statements to be omitted)
  .
#JEND
```

Example: Include statements with dates

This example includes statements from 13200 through 13206.

```
#JI,BD=13200,ED=13206
  .
  . (JCL statements to be included)
  .
#JEND
```

Example: Include statements with several schedule IDs

This example includes statements whenever a job is scheduled by schedule IDs 1, 4, 5, 6, and 9.

```
#JI,ID=(1,4-6,9)
  .
  . (JCL statements to be included)
  .
#JEND
```

Example: Include statements between two times

This example includes statements only between 10:00 a.m. and 4:00 p.m.

```
#JI,OA=1000,OB=1600
  .
  . (JCL statements to be included)
  .
#JEND
```

Example: Omit statements between two times for specific schedule IDs

This example omits statements only between 4:00 p.m. and 10:00 p.m. for schedule IDs 9, 10, and 11.

```
#JO,ID=9-11,OA=1600,OB=2200
  .
  . (JCL statements to be omitted)
  .
#JEND
```

Note: The JCL*xx* text editor command or the LJCK command can simulate the JCL inclusions or omissions. This simulation can help test the conditional JCL statements for proper generation. SASSJCLU can remove expired override statements from PDS libraries.

More information:

SASSJCLU - JCL Utility (see page 328)

Additional Override Statements

CA Workload Automation SE also provides a number of other functions that can be accomplished with statements that also begin with a # in position 1. These statements can be located anywhere within the execution JCL after the JOB statement. (PROCLIB members containing these statements are not seen by CA Workload Automation SE and including them there would thus serve no purpose.) They can be scheduled with #Jx statements if wanted; however, they can be included by themselves if conditional inclusion is not necessary. Only the #MSG statement can be scheduled with the #Xx statement.

These statements are valid only when a job using them is brought into the request queue and not at submit time. Thus, the #X statements cannot be used to schedule them (other than the #MSG statement).

Note: Do not use the following override statements with internal cross-platform job types: #MNT (Maintenance run) and #7UNI (batch cross-platform submission).

The following statements can be placed in the execution JCL:

#ARF

Overrides the ARFSET designation on the job definition panel when the JCL is attached for the job.

#HLD

Places the job in HOLD. Similar to the HOLD command and the job definition panel HOLD option.

#JCL

Sets a JCL override requirement.

#MNT

Considers this run a maintenance type run. Similar to the DB.1 panel MAINT option.

#MSG

Sends the messages to the master station at job submission time.

#NOX

Makes this run nonexecutable. Similar to the job definition panel 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.

#SCC

Defines step-level condition code checking criteria.

#VER

Sets manual verification requirement. Similar to the VERIFY command and the job definition panel VERIFY option.

#7UNI

Flags JCL as a cross-platform submission job. The statement must be the first statement in the member.

More information:

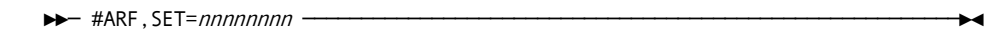
#SCC Statement (see page 322)
#MSG Statement (see page 320)
#RES Statement (see page 321)

#ARF Statement

The #ARF statement lets you override the ARFSET designation on the job definition panel when the JCL is attached for the job.

This statement has the following format:

#ARF



#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.

Example

Suppose that the job definition panel for job A designates an ARFSET named ARECOVR 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 is used to determine ARF recovery for this job instead of ARECOVR.

Usage Notes

When a job initially enters the request queue, CA Workload Automation SE determines whether ARF is to monitor the job. CA Workload Automation SE selects an ARFSET to be used for monitoring and recovery that is based on the following hierarchy:

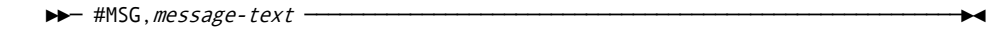
1. If an ARFSET is supplied on the DEMAND or RUN command, it is used.
2. If no ARFSET is supplied on a DEMAND or RUN command for the job, the ARFSET named on the appropriate #ARF statement in the JCL is used.
3. If no ARFSET is supplied on a DEMAND or RUN command for the job, and if no ARFSET is named on a #ARF statement in the JCL, the ARFSET named on the job definition panel is used.
4. If none of the preceding sources supplies an ARFSET reference, ARF does not monitor the job.

#MSG Statement

The #MSG statements let you send messages to the CA Workload Automation SE 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 can appear within scheduled overrides, in which case sending the messages is conditional.

This statement has the following format:

#MSG



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.

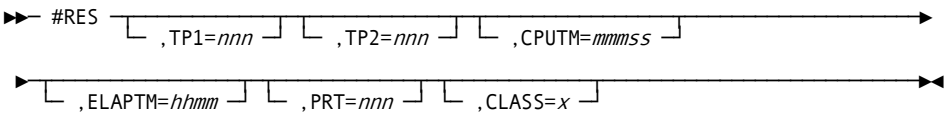
Note: The same rules for other # statements also apply to the #MSG statement.

#RES Statement

The #RES statement in the execution JCL lets you make temporary changes to the high-water workload balancing resource requirements of the job.

This statement has the following format:

#RES



TP1

(Optional) Specifies the number of tape drives of TYPE1 that are required of the job. Not valid for internal cross-platform job types.

Limits: 1 to 3 numeric characters from 0 to 255

TP2

(Optional) Specifies the number of tape drives of TYPE2 that are required of the job. Not valid for internal cross-platform job types.

Limits: 1 to 3 numeric characters from 0 to 255

CPUTM

(Optional) Specifies the amount of CPU time the job consumes in minutes and seconds. Not valid for internal cross-platform job types.

Limits: 2 to 5 numeric characters specified as *mmmss* where *mmm* can be from null to 999; seconds from 00 to 59

ELAPTM

(Optional) Specifies the elapsed time required for the job.

Limits: 2 to 5 numeric characters specified as *hhmm* where *hh* can be from 0 to 24; *mm* from 00 to 59, the highest value being 2400.

PRT

(Optional) Specifies the initial priority value for the job.

Limits: 1 to 3 numeric characters from 0 to 255

CLASS

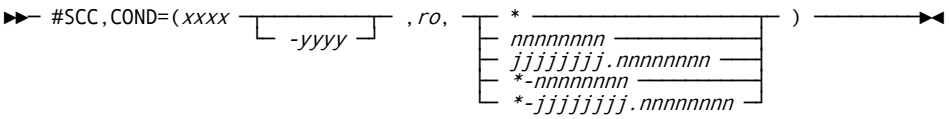
(Optional) Specifies the WLB job class of the job.

Limits: 1 alphanumeric character

#SCC Statement

The #SCC statements in the execution JCL let you define step level condition code tests used to determine whether the job completed successfully.

This statement has the following format:



#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

Specifies the condition code value to test against the actual value returned at execution time. The entry must be numeric and fall within the range of 0 to 9999.

A range of condition codes can be specified as *xxxx-yyyy* (or *yyyy-xxxx*) where *xxxx* is the minimum in the range of condition code values to be tested and *yyyy* 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 9999.

ro

Identifies 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

Specifies greater than.

GE

Specifies greater than or equal to.

EQ

Specifies equal to.

LT

Specifies less than.

LE

Specifies less than or equal to.

NE

Specifies not equal to.

FL

Specifies to 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 jjjjjjjj.nnnnnnnn

(For CPU Jobs) Specifies the step name (and optional procstepname) of the steps to test. 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. To exclude a step from a global test, code 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 *jjjjjjjj* 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 is applied to all occurrences of that job step name or procedure step name.

(For XPJOB Jobs) Only *nnnnnnnn* should be specified, and it should be set to the CCI node name (or alias if present) where the job will execute (that is, the remote node). An * (asterisk) entry indicates that the test applies to all CCI node names. A test that is to be applied to all CCI node names is a global #SCC test. To exclude a CCI name from a global test, code the statement using the following format:

```
#SCC,COND=(xxx,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*.

Agent jobs cannot use #SCC statements because the agent determines success or failure of a job execution.

Example

The following statements are examples of the #SCC statement:

```
#SCC,COND=(16,LT,STEP0030)
```

In the preceding 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 preceding 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.

Usage Notes

- #SCC statements can optionally be scheduled using the #JI or #JO techniques. If the #JI or #JO are used, different #SCC statements can be used per SCHID.
- #SCC statements cannot be scheduled using #XI or #XO and cannot 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) can 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 job definition panel must be set to #S or all #SCC statements are ignored. If job-level condition code testing is wanted, the job definition panel 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 Workload Automation SE 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 can 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 job definition panel is less than using the #SCC statements and should be considered before implementing #SCC tests. For most jobs, the job definition panel 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.

Additional Override Examples

The following statements are more override examples.

Example 1

This example assumes that 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
  .
  .
  .
```

Example 2

This example assumes that 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 (mmmss)       = 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 2 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
  .
  .
  .
```

Example 3

This example assumes that job XYZ runs daily, Monday through Friday. But on Friday, a control statement must be included to indicate a week-ending run and extra 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 ...
//JOBLIB    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
/*
```

Example 4

This example assumes that a job has 10 steps. On STEP01, consider any condition code other than 5 as invalid. If the condition code returned from any step other than STEP01 is not equal to 0, move the job back to the request queue and flag it for restart.

```
#SCC,COND=(5,NE,STEP01)
#SCC,COND=(0,NE,*-STEP01)
```

Example 5

This example assumes that a job has 5 steps. On STEP01 any condition codes other than 0, 8, 16, and 32 should be considered invalid. STEP02-STEP04 has no condition code checking. 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)
```

Example 6

This example assumes that a job has 10 steps. On STEP03, consider any condition code other than 100 as invalid. Only two condition codes are valid for STEP07: 0 and 100. Consider any condition other than 0 as invalid on all remaining job steps. Code the following #SCC statements 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)
```

Example 7

This example assumes that the JOBCHECK step *must* complete with a condition code of zero for the job to complete typically. Code the following #SCC statements:

```
#SCC,COND=(0,NE,JOBCHECK)
#SCC,COND=(,FL,JOBCHECK)
```

More information:

JCL Validation (see page 330)

SASSJCLU - JCL Utility

SASSJCLU is a batch utility program that provides capabilities to perform the following actions:

- Remove expired 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 wanted functions to be performed are indicated through the PARM values to the program. A SYSIN data set can be used to identify particular members to list, update, or both.

The utility requires the following execution JCL:

```
//STEPX     EXEC  PGM=SASSJCLU,PARM='xxxxxxxx.....x'
//STEPLIB   DD    DSN=user-defined-CA-7-loadlib,DISP=SHR
//SYSPRINT  DD    SYSOUT=A
//SYSUT1    DD    DSN=JCL-jcl.dataset.name,DISP=OLD
//SYSIN     DD    *        (optional)
control card              (optional)
/*
```

PARM='*xxxxxxxx....x*'

The acceptable values for PARM='*xxxxxxxx....x*' can be any of the following:

PRINT

Print the JCL members that are selected by control statements in the SYSIN data set.

CLEAN=*nn*

Remove all scheduled JCL overrides from JCL members that are selected for which the ending date (ED) is more than *nn* days before today's date.

PANIC

Remove all scheduled JCL overrides from JCL members selected.

Usage Notes

Usage considerations for PARM options are:

- If PARM is omitted, PRINT is the default.
- CLEAN=*nn* causes CA Workload Automation SE JCL Scheduled Overrides to be removed from the JCL library if their ending date (ED) was more than *nn* days before today's date. If CLEAN is specified, the two-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.

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:

```
1
NAMEKEY=xxxxxxxx
```

where:

xxxxxxxx

Is a JCL member name, up to eight 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 eight characters are used. For example, if a member name is four characters, and they are the same as the first four characters of an eight-character member name, both are considered.

Only one control statement is allowed per each execution of SASSJCLU.

JCL

This illustrates sample JCL for the SASSJCLU utility:

```
//STEPX     EXEC  PGM=SASSJCLU,PARM='PRINT,CLEAN=14'
//STEPLIB   DD    DSN=cai.CAL2LOAD,DISP=SHR
//SYSPRINT  DD    SYSOUT=A
//SYSUT1    DD    DSN=CA7.JCLLIB,DISP=OLD
//SYSIN     DD    *
NAMEKEY=DUSA
/*
```

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.

JCL Validation

The JCL Validation facilities can be invoked from the editor to check for various syntax errors. This validation is available only when the native editor is used in Full Edit Mode (FEM). Thus, if you are using the interface with TSO/ISPF, these subcommands are *not* honored in the ISPF editor.

CA JCLCheck handles the JCL validation. A limited set of CA JCLCheck functions is provided as part of the CA Workload Automation SE base product at no extra cost. However, a more extensive report of errors is available when using the interface with the complete CA JCLCheck.

LOAD Command Processing

Use the LOAD/LOADH command to create or recreate 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.

The RELOAD field of the DB.1 panel serves as an alternative to this command.

Use the LOADH command to indicate that the job is to be entered into the queue in CA Workload Automation SE *hold* status to allow for manual release at a future time.

Note: LOAD processing is not valid with internal cross-platform job types.

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.

Note: For more information about the LOAD/LOADH command, see the *Command Reference Guide*.

More information:

DB.1 - CPU Job Definition Panel (see page 41)

Special Override Library

A supported JCL override library is intended to handle any one-time JCL needs that arise.

The combination of the special override library and the various special override statements let you handle virtually every special need that can arise. See that discussion for more details on the various special functions that are available.

More information:

Scheduled Overrides (see page 311)

Special Override Library Definition

Although PDS, [assign the PANVLT for your book], 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 Workload Automation SE and must be a PDS. Only one special override library can be defined to CA Workload Automation SE.

This special override library must be allocated by the user and made available to CA Workload Automation SE through an appropriate DD statement and an initialization file JCL statement that specifies INDEX=254. This library is then available to the user on the various panels and commands as JCL ID of 254.

Temporary JCL Creation

Execution JCL can be placed here, through the DB.7 panel, any time prior to the scheduling into the queues of the job that uses that JCL. The DB.1 panel field USE-OVRD-LIB must also be set to Y for the special override library JCL to be used. If the DB.7 panel 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 Workload Automation SE 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 job definition panel.

Temporary JCL Usage

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 panel. 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 can be altered using the QM.5 panel if necessary. When the job completes successfully, the JCL is then deleted from the special override library if the member name in the DB.1 panel 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 Workload Automation SE initialization file can be used to prevent this automatic deletion. If this option is set, it applies to all CA Workload Automation SE jobs. For more information, see the OPTIONS statement in the *Systems Programming Guide*. Before using this option, you should read about alternate JCL libraries to see whether they are appropriate for your situation.

More information:

Alternate JCL Libraries (see page 333)

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.

Alternate JCL Libraries

CA Workload Automation SE 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 section, accommodates one-time overrides only and the temporary member is deleted after just one execution of the job. CA Workload Automation SE LOAD processing does not use the alternate JCL library (when using the LOAD command).

Alternate Library Definition

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 can be any of the supported organizations and do not have to be PDS like the global special override library.

Temporary JCL Creation

Temporary JCL can be stored in an alternate library from either CA Workload Automation SE or from outside of CA Workload Automation SE (for example, TSO, CA Roscoe, and so forth). The JCL ID field on the job definition panel still indicates that the JCL resides in the primary library.

Temporary JCL Usage

When CA Workload Automation SE schedules a job that uses a JCL library having an alternate library defined, CA Workload Automation SE searches the alternate library first. If the JCL member is found in the alternate library, CA Workload Automation SE uses the alternate JCL and flags the job queue entry. If the JCL member is not found in the alternate library, CA Workload Automation SE searches the primary library.

Unlike the special override library previously discussed, no action is taken by CA Workload Automation SE 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 panel must specifically request an alternate library since the job definition panel 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 Workload Automation SE 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.

Other Considerations

Unlike the one-time Override library (INDEX=254), which is automatically reset by CA Workload Automation SE, 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: Using the Edit Facility

This section contains the following topics:

Overview (see page 335)
Active Area (see page 336)
Edit Work File (see page 337)
Environments (see page 337)
Editor Usage (see page 337)
CA 7 Text Editor Environment (see page 338)
TSO/ISPF Editor Environment (see page 361)

Overview

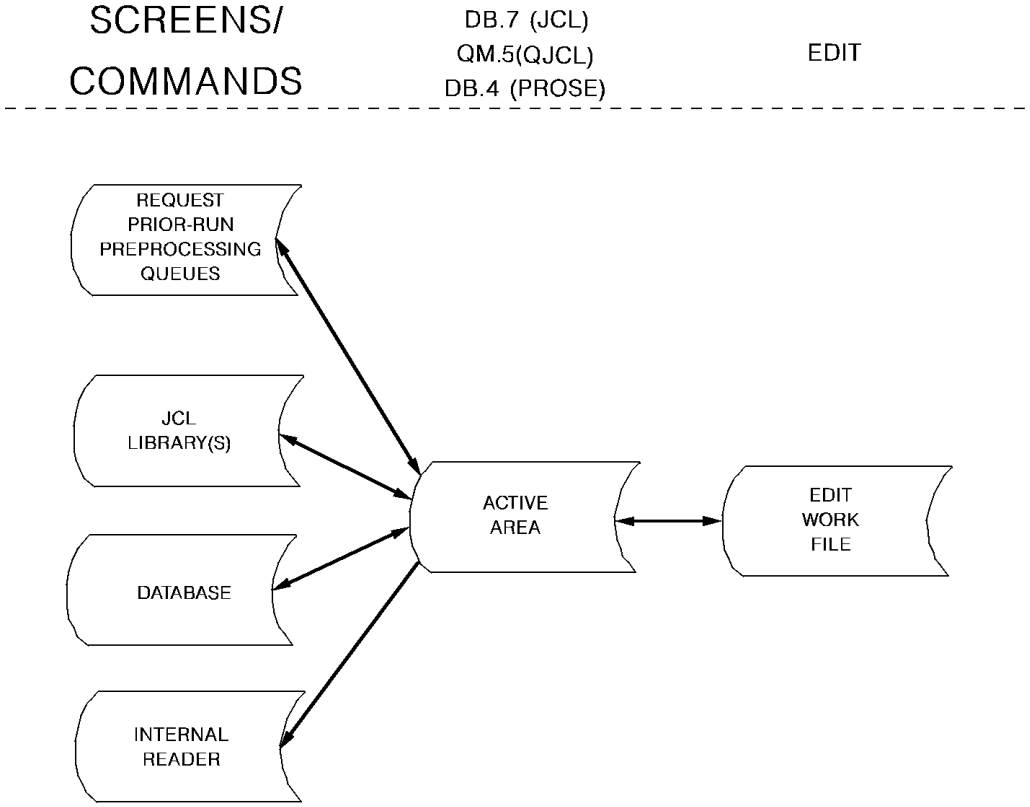
The CA 7 edit facility is an interactive component that can create, modify, and manage 80-character card-image data. With this facility, job streams (JCL and PARM data) can be created or changed before submittal and documentation (prose) can be maintained in the database.

This facility consists of files, panels, programs, and a command. The command is EDIT that can be issued as a top line command or as a panel function. The files are referred to as active area and the edit work file (EWF). The panels and programs vary based on the environment from which EDIT is issued.

When this facility is invoked, the active area can be cleared or loaded with source data, when requested. The active area is copied to the EWF and an editor program is executed to make the wanted changes. When editing is complete, any saved data is copied back to the active area. From the active area, it can be copied to the source file or some other target file.

Note: For CA7TOUNI jobs and for any internal cross-platform jobs, the editing session always appears in mixed case mode, allowing both uppercase and lowercase letters.

The following figure shows the edit facility data flow.



Active Area

The active area is a dynamic file that 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.

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 panels show a field that is named ACTIVE SIZE. The field tells the terminal user how many lines of text are in this active area. A CLEAR function is performed at every logon and logoff.

Edit Work File

The edit work file (EWF) is a dynamic file that is created for each terminal user when EDIT is invoked. For the CA Workload Automation SE 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 environment of the terminal user to change the EWF data. When editing is complete, any saved updates are copied back to the active area and the EWF is freed.

Environments

Currently, the edit facility supports the CA Workload Automation SE and TSO/ISPF environments. Although the editing process is similar, the editor programs that are used in each environment are different. Also, areas unique to each environment are in this section.

Editor Usage

Using the editor involves the following three general activities:

- Invoking the editor
- Editing text data
- Leaving the editor

When using CA Workload Automation SE, the editor is invoked from another command or panel, such as DB.7 or DB.4, although it can be requested with the EDIT top line command. While under the control of the editor, several subcommands are available for editing the text data. After making any wanted changes, it is necessary to leave or exit from the editor to retain or use the updated text.

Invoking the Editor

Any of the following methods can invoke the editor:

- Using the top line EDIT command.
- Using the EDIT or FE (Fetch and Edit) functions from database maintenance or queue maintenance panels or commands.

These functions cause the transfer of data from the active area to an edit work file, as described previously. Next, they engage the editor programs for the appropriate environment.

Editing Text Data

Once in the editor, text can be created or changed using the subcommands appropriate to the environment where editing occurs.

Leaving the Editor

After completing the editing tasks wanted, the editor must be exited and changes saved, if wanted. To save changes, the editor must be told to retain the updates, and the updated EWF data must be transferred back to the active area. To cancel the changes, the editor must be told to release the EWF and not transfer data back to the active area.

For the CA Workload Automation SE 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.

For the TSO/ISPF environment, ISPF edit macros are provided. They correspond to required CA 7 editor subcommands. Also, when the editor is invoked, a default subcommand is set. If cancel the changes, request the CA7EXIT subcommand.

More information:

CA 7 Text Editor Environment (see page 338)

CA 7 Text Editor Environment

The CA 7 text editor routines provide the capability to create or change text information in the database, queues, and JCL data sets. These routines operate in the CA Workload Automation SE 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 panels or batch card-image input.

Edit Modes

Earlier versions of CA Workload Automation SE 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 Panel or from any formatted panel (other than a DB.2.n panel), 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.

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 panel where EDIT is *not* a listed function (for example, DB.1, DB.6), return is to the DB Menu panel.
- If the text editor was entered from a formatted panel (other than DB.2.n) where EDIT is a valid function (for example, DB.7, DB.4), the return is to the originating panel.
- Unprintable (or hex) data can cause a 3270 screen error and terminate the user session.
- If you do not have a data set assigned to JCL-ID zero, some of the FEM subcommands do not yield the intended results. For example, in this situation, the SS subcommand does not save your changes into the intended library, only in the edit work area. You must then enter the JCL command to invoke the CA-7 JCL Library Maintenance panel and then enter SAVE as a function while supplying the member name and a valid JCL-ID.

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.

A Full Edit Mode (FEM) panel 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

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 two. 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 panel, the command line, appears on the left half of line two. 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 panel display line. When an error occurs or specific status is required, this line is used. Under normal processing, this line remains blank.

The sixth area on the FEM panel 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 the data area has no changes, 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 the data area has changes, 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 panel, 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 no other DDs are coded deletes all lines on the current page from the EWF once Enter is pressed.

I*nnn*

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 can 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.

R*nnn*

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 I*nnn*.

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.)

***BOT*TOM**

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.

***C*OPY *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 can be renumbered (using the current system increment) to ensure that the EWF remains in sequence. Available in FEM and AEM.

***D*ELETE 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.

***E*DIT**

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.

***E*DIT LIST [*m*,*n*]**

Enters Edit List mode. Lines displayed can be changed by keying the changes into the EWF and pressing Enter. A line displayed on the panel can be deleted by keying $EDEL in the first five characters, or by positioning the cursor at the first character of the line and pressing the EOF key (erase to End-of-Field). Panels are returned until either the end of the EWF is reached, line *n* is displayed, or $EEND is keyed on the last line. ($EEND can 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.)

***E*DIT /*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.

***E*DIT /*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.

***E*DIT /*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.

***E*DIT //*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.

***E*DIT *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:

This example changes scan columns to be columns 73 through 80.

```
EDIT 73,80
```

This example fills columns 73 through 80 with SASSEDIT.

```
'E //SASSEDIT/'
```

This example resets scan columns.

```
E 1,72
```

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, 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.

The following example 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.

```
FIND /ABC/0,99999
```

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.

***I*NSERT [*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 panel 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 can 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 Workload Automation SE 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) can 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 Workload Automation SE 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) can 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 Workload Automation SE 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) can 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 Workload Automation SE 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) can 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 Workload Automation SE 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) can be overridden using *yyddd*. Use *hhmm* to override the default time (current time).

***L*IST [*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 uppercase and lowercase characters to be entered. For more information about changing the case setting, see the discussion of the UPPER command in this section.

MIXED is always the default setting for CA7TOUNI jobs and for all internal cross-platform jobs, regardless of any options set and data contained in the file. With CPU jobs, other than CA7TOUNI jobs, the following applies:

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 Workload Automation SE 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 Workload Automation SE initialization file.

***M*OVE *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 can 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*

Renumbers 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 panel that was used to enter the text editor, and a REPL function is performed. Available in FEM and AEM.

Note: If JCL or PARM data is being replaced in the request queue, the SR command also sets an outstanding JCL or PARM data override requirement off.

SS

Performs two functions, SAVE and SAVE. The current EWF replaces the active area, control returns to the panel 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. Press Enter to redisplay the EDIT panel 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 Workload Automation SE initialization file then the initial case setting is 'MIXED'.

Note: For CA7TOUNI CPU jobs and for all internal cross-platform jobs, the initial setting is always MIXED. You can specify UPPER if you want to have all data translated to uppercase characters.

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 Panels

The following are FEM sample panels. Each panel displays the use of subcommands in the sequence field areas or changes made to the data area.

The following panel shows five of the six areas discussed previously. The last area of the panel, the message line, is blank.

FEM Sample Panel 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 panel reflects a data area change on line 60 of FEM Sample Panel 1. Make changes by keying over existing line data and then pressing Enter.

FEM Sample Panel 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 panel reflects the request for deletion of three lines from the EWF. You can place a D in any position of the sequence field to delete the line.

Note: For the results of this edit, see FEM Sample Panel 4.

FEM Sample Panel 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,*************************************************************************
D0080 #JI,ID=2
00090 #MSG,*************************************************************************
00100 #MSG,*              NETWORK RUN AS SCHID OF 2                                *
00110 #MSG,*************************************************************************
0012D  #JI,ID=3
00130 #MSG,*************************************************************************
00140 #MSG,*              NETWORK RUN AS SCHID OF 3                                *
00150 #MSG,*************************************************************************
00D60 #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 panel reflects the results of the edit performed in FEM Sample Panel 3 deleting three lines: 00080, 00120, and 00160.

Now a P is placed in the sequence field on line 100.

Note: For the results, see FEM Sample Panel 5.

FEM Sample Panel 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 panel reflects the result of the P subcommand to position the display from FEM Sample Panel 4 to line 100.

Now an I3 is placed in sequence line 200.

Note: For the results, see FEM Sample Panel 6.

FEM Sample Panel 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=CA7.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=user-defined-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 panel 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. For the results, see FEM Sample Panel 7.

FEM Sample Panel 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=CA7.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 panel 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 Panel 6. Lines 150 through 200 are deleted from the EWF.

FEM Sample Panel 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=CA7.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=user-defined-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 panel reflects multiple subcommands entered on the panel.

FEM Sample Panel 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=CA7.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=user-defined-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 panel reflects the results of multiple subcommands entered on the panel displayed in FEM Sample Panel 8. 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 Panel 9

```
      ----+----10---+----20---+----30---+----40---+----50---+----60---+----70*
PF                                                NOFILL I(010)
00260 //SYSUT2 DD DSN=CA7.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=CA7.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*
```

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 panel is the DB.7 panel, but the same steps apply to other panels, such as DB.4 and QM.5.

See the figure at the beginning of this section.

To update text, proceed as follows:

1. Display the primary panel, in this case, DB.7.
2. Type FE in the function field, and 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 wanted changes to the text.
4. Enter the SAVE or SR subcommand to retain the changes and return to the DB.7 panel. If SAVE is used to leave the editor, another function is required on the DB.7 panel 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.

Creating Text

Another common editing activity is to create text from scratch. The following example is a step-by-step example for this activity using the CA 7 text editor. In the example, the primary panel is the DB.4 panel, but the same steps apply to other panels, such as DB.7 and QM.5.

See the figure at the beginning of this section.

To create text, proceed as follows:

1. Display the primary panel, in this case, DB.4.
2. Since this panel is a menu panel, a secondary entry is required, such as 1 for job level documentation.
3. Once on the DB.4.1 panel, 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 panel that is returned indicates EWF EMPTY.
5. Enter I or INSERT that invokes Data Insert processing.
6. Enter text as wanted. Lines are double spaced with sequence numbers protected. If text is entered on all displayed numbered lines, another panel of blank lines is presented for continued entry. After all input is complete, leave at least one blank line on the panel and press Enter to return to FEM.
7. If any 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 panel.
9. Type SAVE in the function field and the job name in the JOB field. Data can be entered in the other fields too. Press Enter to add the text to the database.

   Note: If you are entering no data in the other fields, use SS in Step 8. SS performs steps 8 and 9 with one command.

Special Considerations

This topic explains special considerations for using PF/PA keys, character translation, and nondisplayable data (hex).

PF/PA Key Usage

While in the CA 7 text editor, PF and PA keys should not be used. These keys can 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 because the EWF is not saved to the Active Area.

Character Translation

CA Workload Automation SE translates all online character data to uppercase unless the text editor is used in Full Edit Mode. If you use FEM, character data is translated or not depending on the case setting. If the case setting is UPPER, all character data are translated to uppercase. If the case setting is MIXED, characters on data lines are left untranslated. The FEM subcommands MIXED and UPPER can be used to change the case setting. The status line reports the current case setting.

On entry to the CA 7 editor in Full Edit Mode (FEM) for CA7TOUNI CPU jobs and for all internal cross-platform jobs, the default case setting is always MIXED, regardless of the options set or the data contained in the text being edited. You can change this setting to 'UPPER' if you want only uppercase data.

On entry to the CA 7 editor in Full Edit Mode, the case setting is UPPER unless lowercase characters are detected. In that event, the initial case setting is MIXED.

If the case setting is changed to UPPER,all character data in the editor is translated to uppercase when it is saved to the active area. If data is not saved, the case setting only affects the current data display.

The case setting cannot be changed unless INITCASE=Y is specified on the OPTIONS statement in the initialization file. If this value is not specified, 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.

Nondisplayable Data (hex)

Nondisplayable data that is contained in text that is edited using the CA 7 text editor can cause unpredictable panel output or a terminal disconnect. Avoid editing files containing such data.

TSO/ISPF Editor Environment

The CA Workload Automation SE TSO/ISPF Interface allows for the fully integrated use of the ISPF editor where text editing is required. Some of the previously described features are available in the ISPF environment too. Some important differences exist, and these differences are noted in this subtopic.

This subtopic explains the use of the ISPF editor only as supported under the CA Workload Automation SE TSO/ISPF Interface.

In the following explanations, two types of CA Workload Automation SE online terminal sessions are compared and contrasted:

- A *directly connected* CA Workload Automation SE terminal session
- An *ISPF connected* CA Workload Automation SE terminal session

A CA Workload Automation SE online terminal session that is acquired outside of ISPF is considered *directly connected*. A CA Workload Automation SE terminal session that is 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 Workload Automation SE 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 the 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 Workload Automation SE. CA Workload Automation SE handles the manipulation of data on the file entirely in its address space in response to CA 7 editor commands that the user entered.

In an *ISPF connected* session (in a CA Workload Automation SE terminal session acquired under ISPF), the EWF is a temporary data set that is dynamically allocated and is 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 Workload Automation SE from the CA Workload Automation SE 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.

Requesting Edit Functions from the ISPF Editor

Several CA Workload Automation SE functions can be requested from the editor panels that 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 Workload Automation SE terminal session, the following subcommands are used to request these functions:

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 Workload Automation SE; therefore, some means must be provided to allow the user to indicate to the interface programs from the editor, what CA Workload Automation SE is to do with the data in the EWF when the edit session is terminated. In an *ISPF connected* CA Workload Automation SE terminal session, the following ISPF edit macros can be used to request the CA Workload Automation SE edit functions described previously:

CA7SAVE

Terminates the edit session. Sends EWF to CA Workload Automation SE and requests SAVE processing. Deletes the temporary data set (EWF).

CA7EXIT

Terminates the edit session. Sends EXIT request to CA Workload Automation SE. Deletes the temporary data set (EWF).

CA7SS

Terminates the edit session. Sends EWF to CA Workload Automation SE and requests SS processing. Deletes the temporary data set (EWF).

CA7SR

Terminates the edit session. Sends EWF to CA Workload Automation SE 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 Workload Automation SE edit function (for example, SAVE, SS, SR, or EXIT).

Default Requests for CA Workload Automation SE Edit Functions

In an *ISPF connected* CA Workload Automation SE terminal session, a CA 7 editor function is always requested when the editor is exited. This function can be requested either explicitly, as in the case of one of the edit macros. Also, the function can be requested by default if the editor is exited in any other way. 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

xxxx can be either EXIT, SAVE, SS, or SR. The default action is set to SR initially. The default setting can 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, on the next session, the default setting that appears in the message is SS. This is the action that CA Workload Automation SE 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 Workload Automation SE edit macro, a CA Workload Automation SE 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 Workload Automation SE edit macro. AUTOSAVE ON is a common edit profile setting and in many cases data can be saved automatically if data is changed. It is very important to be aware of the default CA Workload Automation SE edit function that can 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 panel 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), and press Enter.
3. The ISPF editor is invoked and data from the active area appears. Use ISPF editor commands to make wanted 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 Workload Automation SE edit function is acceptable and if data was changed and if AUTOSAVE ON is set in the edit profile, simply exit the editor using the ISPF END command (through a PF KEY if wanted).

Special Considerations

This topic explains special considerations for using ISPF edit profile settings, PF/PA keys, the SUBMIT function, and the size of data.

ISPF Edit Profile Settings

In an *ISPF connected* CA Workload Automation SE 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 because CA Workload Automation SE is handling all output to the primary source/target. When the EWF is received, CA Workload Automation SE 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 can be used to create unprintable hex data, this is strongly discouraged because the response of CA Workload Automation SE to such data is unpredictable.

PF/PA Key Usage

Although both CA Workload Automation SE and ISPF allow command input through a PF key, in the ISPF editor all PF keys are used for ISPF command input only.

Note: For more information about PF keys in an *ISPF connected* CA Workload Automation SE terminal session., see the *Systems Programming Guide*.

The use of PA keys is strictly subject to ISPF restrictions. PA key interrupts are not supported in an *ISPF connected* terminal session.

SUBMIT Function

If JCL is submitted using the ISPF SUBMIT function, CA Workload Automation SE *does not track* the job. We recommend that the SUBMIT function be deactivated through an ISPF command table for the CA7 application.

Note: For more information about command table modifications for the CA7 application, see the *Systems Programming Guide*.

Size of Data

The maximum number of lines (or records) that can be edited using the ISPF interface is approximately 5000.

Chapter 12: Internal Cross-Platform

Three formatted panels permit entry of data used explicitly by internal cross-platform jobs during the submission process. The first two panels are for XPJOBs, and the third panel is for agent jobs. Although the panels are not accessed through any menus, you can use the panels' IDs (XN.*n* where *n* is 1, 2, or 3) as top line commands or enter them through the DBM panel field FUNCTION. The command aliases for these panels are XNODE, XPSWD, and AGPSWD.

This section contains the following topics:

XN.1 - XNODE Command (see page 367)
XN.2 - XPSWD Command (see page 370)
XN.3 - AGPSWD Command (see page 375)

XN.1 - XNODE Command

The XN.1 or XNODE command lets you define permanent nodes used by XPJOB jobs.

The CAICCI nodes to which internally submitted XPJOB jobs are routed for execution are managed through an in-storage table. This table is initially built from the NODE records, stored in the CA Workload Automation SE VRM database. The database node records contain the node names (primary, alternates 1 and 2), and the time of last command update if applicable. These nodes are classified as permanent nodes. The data associated with nodes in the in-storage table also includes the time of last transmission to the node or its alternates, as well as the number of jobs transmitted since the last time CA Workload Automation SE Online was started.

If a node is not in the CA Workload Automation SE database but is in the in-storage copy of the node table, it is classified as a temporary node. These node entries are added dynamically if CA Workload Automation SE sends a job to a node that is not previously defined. These temporary nodes can be added to the database if an XNODE ADD function is executed.

To display the panel, enter:

- XN.1 as the function on a DBM panel or the top line command.
- XNODE as the function on a DBM panel or the top line command.

To exit the panel:

- Press PF3 to return to the DB Menu panel except with the TSO/ISPF interface when PF3 has been assigned to another function.
- Move the cursor to the top line of the panel and enter a top line command for another function.

```
------------------------  CA-7 XPS Node Table Entry ------------------------
Function: xxxxxxxx  (Add, Delete, Format, List, Update)

Node: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

 Description: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

 Alternate 1: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

 Alternate 2: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

 State:       xxxxxxx                 (Online, Offline, Stop)









 PROGRAM: SMX1  MSG-INDX: nn   -- XN.1      --   yy.ddd / hh:mm:ss
 MESSAGE: Enter function, transfer or enter a command on the top line
```

Function

Specifies the function to perform. Value must be the name of some other panel or one of the following:

ADD

Adds a new primary node definition to the active node table and to the database. The ADD function does not perform updates against any existing permanent node definition. Use the UPDATE function instead. It can be used to make a temporary node entry a permanent node entry (thus adding it to the database).

DELETE

Deletes an entry in the active node table and the database. This removes the primary node entry.

FORMAT

Clears all fields on the XNODE panel.

LIST

Lists an entry in the active node table at the time of the command.

UPDATE

Updates the alternate node definitions, status information, or both.

Batch keyword: Positional parameter

Node

Defines the primary node (the *key*) in the function being performed. This keyword is required for all functions.

Limits: 1 to 44 alphanumeric characters. Although 44 characters are permitted in this field, currently CA Common Services CAICCI only permits 8-character node names.

Batch keyword: Positional parameter

Description

(Optional) Provides a 64-byte field for sites to describe the node. This field can indicate anything a user wants and is not used in any CA Workload Automation SE data. Suggestions for this field include the operating system (for example, UNIX, Windows), location (for example, New York, Dallas), and other data relevant to the platform.

Limits: 1 to 64 alphanumeric characters

Batch keyword: DES

Alternate 1|Alternate 2

(Optional) Names the alternate CAICCI nodes to use if the primary node is unavailable. ALT1 must be specified before an ALT2 is used. Users may see an error message if ALT2 is specified when the primary node does not have an ALT1 defined.

Even though a CAICCI node can be assigned as an alternate, that same CAICCI node can also be used in a primary definition. CA Workload Automation SE only uses the primary node's alternates if the primary node is unavailable at the time the job is submitted.

For example, NODE-A is defined with NODE-Z as its alternate. NODE-Z also has a primary node definition, with NODE-G as its alternate. A job destined for NODE-A when NODE-A is not available is routed to NODE-Z if possible. If NODE-Z is unavailable, the job waits in the request queue until NODE-A or NODE-Z becomes available. The job is never routed to NODE-G unless NODE-G is added as an alternate to NODE-A.

Limits: 1 to 44 alphanumeric characters

Batch Keyword: ALT1 or ALT2

State

(Optional) Changes the status of the node from online (active) to offline (inactive) or stop. Offline states that only this node should not be used to send any data; alternate nodes may be used. Stop status indicates that the primary and its alternate nodes should not be used at all; no data sends will occur to the primary or alternate nodes.

If it is known that the node is unavailable for a maintenance window or extended period of time, the operator can enter this command as the maintenance window begins. This prevents CA Workload Automation SE from issuing messages if it detects the node is offline when a job is submitted for that node.

Note: The status can appear as FAIL if the CAICCI connection is determined by CA Workload Automation SE not to be active at the time of an XPJOB submission.

Limits: 1 to 7 characters

Batch Keyword: STATE

XN.2 - XPSWD Command

The XN.2 or XPSWD command displays a formatted panel to permit entry of security-related data for XPJOB jobs. The panel is a secured panel, which means that access to the panel and the information about it is controlled.

Note: For more information, see the Security *Reference Guide.*

If the XPDEF file initialization statement XSUBMIT keyword is set to N, the panel is disabled for all users.

The XPSWD panel allows the definition of the user ID and optionally, the password and Windows Domain name that an XPJOB job should use when building the request for transmission. The Owner or Node field on the XPSWD panel information should match the Owner or Node field in the actual XPJOB job definition (DB.10). Because multiple XPJOB definitions can have the same Owner or Node, the XPSWD panel eliminates the need to associate user-ID, password, and domain information with each XPJOB definition.

The XPDEF initialization file statement PSWDLOC keyword determines the order in which the XPSWD information is checked when building the request. Consider the following setting:

XPDEF PSWDLOC=(DATABASE,NODE)

With this setting, CA Workload Automation SE looks at the XPSWD OWNER access records for user-ID, password, and domain information before looking at XPSWD NODE access records.

Note: For more information about this search hierarchy, see the *Systems Programming Guide*.

For example, assume that XPJOB XYZJOBA is defined to the CA Workload Automation SE database and its OWNER field is XUSERA. Using the XPSWD panel, you can create an XUSERA OWNER access record where the user-ID is xuseraxpjobfromca7, and xuserapswd is the password. When job XYZJOBA is requested, it includes xuseraxpjobfromca7 and xuserapswd in the request it transmits to the target system. In fact, any XPJOB definition that has an OWNER of XUSERA uses this same information in the request it transmits to the target system.

Now, assume that XPJOB XYZJOBB has its OWNER field set to XUSERB and its NODE is XNODEB. Using the XPSWD panel ,you create an XNODEB NODE access record where the user-ID is xnodebxpjobfromca7 with no password or domain. Note that there is no XUSERB OWNER record. When job XYZJOBB is requested, it includes xnodebxpjobfromca7 in the request it transmits to target system XNODEB. As above, all XPJOB jobs that are routed to XNODEB would use this same information (assuming no OWNER access record matched the OWNER field in the XPJOB definition).

OWNER or NODE can be entered on the XPSWD panel but not both simultaneously.

The XPDEF PSWDLOC value can also be set to USER. When this value is used, the input PARM data for the XPJOB job can be the source of the user-id, password, and domain information.

Always in CA Workload Automation SE, the password is in nondisplay format. If a request is to display the user ID and password, the password field is blank.

To display the panel, enter:

- XN.2 as the function on a DBM panel or the top line command.
- XPSWD as the function on a DBM panel or the top line command.

To exit the panel:

- Press PF3 to return to the DB Menu panel except with the TSO/ISPF interface when PF3 has been assigned to another function.
- Move the cursor to the top line of the panel and enter a top line command for another function.

```
------------------- CA-7 XPJOB User ID/Password Definition -------------------
Function: xxxxxxxx  (Add, Delete, Format, List, Update)

   Owner: xxxxxxxx
or  Node: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

      User ID: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

      Password:                              (Password Not Present)

      Reconfirm PSWD:

      Domain Name: xxxxxxxxxxxxxxx           (Windows platform)









 PROGRAM: SMX2  MSG-INDX: nn   -- XN.2      --   yy.ddd / hh:mm:ss
 MESSAGE: Enter function, transfer or enter a command on the top line
```

XPSWD

Indicates that the password command is entered, and a positional parameter.

Function

Specifies the function to perform. Value must be the name of some other panel or one of the following values:

Add

Adds a user/password for an owner or node to the database when the same named entity (owner or node) does not already exist.

Delete

Deletes an owner/node entity and its associated data from the database.

Format

Clears the panel of all input data.

List

Lists an owner or node entity. In batch, a formatted panel is not listed; only a found or not found message is returned.

Update

Updates database information about an owner or node entity.

Batch Keyword: Positional parameter

OWNER

Indicates to create an owner record. This owner equates to the owner field of the XPJOB job panel, so that multiple jobs all use the same user ID and optional password/domain name. Enter OWNER or NODE, not both.

Limits: 1 to 8 alphanumeric characters

Batch keyword: OWNER

NODE

Defines a default user ID and password to use when PSWDLOC=NODE is coded and used through the XPDEF PSWDLOC security hierarchy. The node must exist as a permanent entry in the node table. Enter OWNER or NODE, not both.

Limits: 1 to 44 alphanumeric characters

Batch keyword: NODE

User ID

Defines the user ID to insert into the information sent to the alternate platform with a job for execution. If a user specifies the ROOT user ID, a verification is made against the XPDEF SUBROOT setting. If SUBROOT is set to NO, an error message is displayed if root or ROOT is entered for the user ID.

Limits: 1 to 32 alphanumeric characters

Batch keyword: XPUSER

Password

(Optional) Defines the password associated with the user ID to insert into the information sent to the alternate platform with a job for execution.

Note: In an online session, this is a non-displayable field.

Limits: 1 to 14 alphanumeric characters

Batch keyword: XPPSWD

Reconfirm PSWD

(Optional) Defines the password that is associated with the user ID, and this password should match the PASSWORD field previously entered. If a match does not occur, an error message appears asking for validation.

Note: In an online session, this field does not display.

Limits: 1 to 14 alphanumeric characters

Batch keyword: RXPPSWD

Domain Name

(Optional) Defines the domain name if the user ID and password are targeted for a Windows (NT) platform. Windows permits multiple domains to be executing in one machine, and this qualifies the user ID and password targeted for a specific domain on a Windows operating system.

Limits: 1 to 15 alphanumeric characters

Batch keyword: XPDMN

(Password Present)/(Password Not Present) Message

The XPSWD panel displays the (Password Present)/(Password Not Present) message to indicate the presence or absence of a password. The SASSHIS8 Transaction Detail report (Report 02) and the Database Update Transaction Detail report (Report 12) do not display the message.

The meaning of the message depends on the success of the function.

- When successfully adding or updating, the message indicates whether a password is associated with the record in the database.
- If an add function is unsuccessful (an error message is received at the bottom of the panel), the Password Not Present message indicates that you must reenter this information to associate a password with the record.

  Note: A password is *not* required.

- If an update function is unsuccessful (an error message is received at the bottom of the panel), the Password Not Present message indicates that you must reenter this information to update password information for this record. The information already in the database is *not* updated unless the update completes successfully.
- When the list function is successful, the message indicates the presence or absence of a password for the record in the database.
- The delete and format functions always display the Password Not Present message.

You have two options if you want to remove a password that is associated to an Owner or Node Password record:

- Delete the record and reenter it without a password.
- Alternately, blank out the password using the Update function, which resets the password to a null value.

XN.3 - AGPSWD Command

The XN.3 or AGPSWD command displays a formatted panel to permit entry of security-related data for agent jobs. Because this is a secured panel, access to the panel and the information on it is controlled. If the XPDEF file initialization statement AGENTJOB keyword is set to NO, the panel is disabled for all users.

Note: For more information about secured panels, see the *Security Reference Guide*.

For agent jobs that require a password, the user ID, agent, job type, and optionally a source field are used as a hierarchical key at job submission time to search a password database. The password database supports specific entries for user ID and specific or generic entries for agent, job type, and source. The search returns the most specific match.

The AGPSWD panel allows for the definition, update, deletion, or existence check of passwords associated with a hierarchical key composed of user ID, agent, job type, and source. The source field is associated with a limited number of job types. The user ID is a required field, and agent, job type, and source are optional fields. Wildcard characters are not permitted in any of the four key fields. When adding or updating a password record, if no data is supplied for an optional field (agent, job type, or source), that field is treated like a generic * mask and matches all input values during the job submission password search process.

The password is never shown on the panel, and the LIST option only returns a message indicating whether the password record was found.

To display the panel, enter:

- XN.3 as the function on a DBM panel or the top line command.
- AGPSWD as the function on a DBM panel or the top line command.

To exit the panel:

- Press PF3 to return to the DB Menu panel except with the TSO/ISPF interface when PF3 has been assigned to another function.
- Move the cursor to the top line of the panel and enter a top line command for another function.

```
------------------- CA-7 Agent User ID/Password Maintenance ------------------
Function:           (Add, Delete, Format, List, Update)


User Id:

Agent:

Job Type:

Source:

Old Password:                                    (for Update only)

New Password:

Verify Password:



PROGRAM: SMX3   MSG-INDX: 00   -- XN.3     --   yy.ddd / hh:mm:ss
MESSAGE: Enter function, transfer or enter a command on the top line
```

This panel contains the following fields:

Function

Specifies the function to perform. Value must be the name of some other panel or one of the following values:

Add

Adds a password record to the database when the same named entity does not already exist.

Delete

Deletes a password record from the database.

Format

Clears the panel of all input data.

List

Verifies the existence of a password record through a displayed message. In batch, a formatted panel is not listed; only a found or not found message is returned.

Update

Updates a password record on the database.

Batch keyword: Positional parameter

User Id

Indicates a user ID that is passed with a password to an agent for authorization.

Limits: Two 64-byte lines composing 128 characters. The wildcard characters * and ? are not permitted.

Batch keyword: AG1, AG2

Agent

(Optional) Indicates the agent that is the target of a job submission and where the user ID and password authorization takes place. If omitted on an add or update, it indicates that this password record is eligible for selection for all agent values.

Limits: 1 to 16 characters. The wildcard characters * and ? are not permitted.

Batch keyword: AGENT

Job Type

(Optional) Indicates the job type of the job that requires a user ID and password authorization. If omitted on an add or update, it indicates that this password record is eligible for selection for all job types.

Limits: 1 to 16 characters. The wildcard characters * and ? are not permitted.

Batch keyword: AGJOBTYP

Source

(Optional) Indicates the source field for a job that requires a user ID and password authorization. This field applies to a limited number of job types. If omitted on an add or update, it indicates that this password record is eligible for selection for all source values.

Limits: 1 to 64 characters. The wildcard characters * and ? are not permitted.

Batch keyword: AGSOURCE

Old Password

(Optional) Defines the current password that is associated with the four key fields (user ID, agent, job type, and source).

Note: This field does not display.

Limits: 1 to 32 characters. Used only on an UPDATE request.

Batch keyword: AGOLDPW

New Password

(Optional) Defines the new password to associate with the four key fields (user ID, agent, job type, and source).

Note: This field does not display.

Limits: 1 to 32 characters. Used only on an ADD or UPDATE request.

Batch keyword: AGNEWPW

Verify Password

(Optional) Defines the new password again for confirmation purposes. If a match with New Password does not occur, the panel displays an error message.

Note: This field does not display.

Limits: 1 to 32 characters. Used only on an ADD or UPDATE request.

Batch keyword: AGVERPW

Chapter 13: 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 Workload Automation SE. Analyze Commands are used to list, and optionally update, database contents from CA Workload Automation SE terminals.

This section contains the following topics:

Pointer Verification (see page 379)
Analyze Commands (see page 384)

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.

Pointer Verification Parameters

The PARM or DBVPARM DD statement parameters available to request pointer verification are as follows:

IDSVERFY

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 Workload Automation SE 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 Workload Automation SE database. ALLVSAM indicates the database is totally VSAM as defined in the control statements residing in the DBPARMS data set.

Note: For details on the contents of the DBPARMS data set, see the *Systems Programming Guide*.

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[,IDSVERFY][,NWSCHD][,JOBSCHD]'
```

-or-

```
//DBVPARM DD *
ALLVSAM [,IDSVERFY][,NWSCHD][,JOBSCHD]
```

When using the DBVPARM DD, the parameters must start in column 1.

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=user-defined-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
//*
//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.

Pointer Verification by Record Type

The following shows how pointer verification proceeds by record type, the keyword that 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 *Message Reference Guide*.

Network Directory

IDSVERFY 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).

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.

Data Set Directory

IDSVERFY keyword:

Checks database index entry for the triggering data set's (AUTO.D*znnnnnn* where *z* is either the character S or a number) index pointer to the Dataset entry, and DSNBR correspondence between member and directory.

Issues error messages 059 through 062.

Documentation Directory

IDSVERFY keyword:

Checks appropriate database index pointer for PP.*network*.NW, PP.*jobname*.JOB, and PP.*system*.SYS.

Issues error message 055.

Job Directory

IDSVERFY 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.

Data Set Member

IDSVERFY keyword:

Checks database index entry for the triggering data set's (AUTO.D*znnnnnn* where *z* is either the character S or a number) index pointer to Dataset entry, and DSNBR correspondence between member and directory.

Issues error messages 059 through 062.

Network Member

IDSVERFY keyword:

Checks index entry for network triggers (AUTO.NW*nnnnnn*) and index NW. entry pointing to network name.

Issues error messages 063 and 064.

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.

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.

Documentation Member

IDSVERFY 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.

Job Member

JOBSCHD,IDSVERFY keywords:

Checks that the first record of a job member is a job entry.

Issues error message 011.

Analyze Commands

These commands are used to request an analysis and to list certain information from the CA Workload Automation SE database and queues. Options are also provided that 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 jobs

RQMT

Predecessor job references

RQVER

Unsatisfied internal JOB/DSN requirements for request queue jobs

TRIG

Triggered jobs cross reference

XREF

Using-job references

The *Command Reference Guide* discusses each command in detail. These commands can 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, we recommend 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 Workload Automation SE low processing activity.

BACKUP, RELOAD, and database verification utilities perform additional database checking. This chapter provides database verification utility information.

Note: For information about the BACKUP and RELOAD utilities, see the *Systems Programming Guide*.

Chapter 14: Database Transportability

This section contains the following topics:

Overview (see page 387)
Assumptions (see page 389)
Database Extraction (see page 395)
Moving Jobs (see page 399)
Workstation Networks (see page 406)
Application Data Sets (see page 408)
PDS JCL Members (see page 410)
CA Librarian JCL (see page 412)
CA Panvalet JCL (see page 415)
Cataloged Procedures (see page 418)
Documentation (see page 420)
Schedules (see page 422)
Node and Access File (see page 427)
Agent DIV File (see page 428)
Create Command Files (see page 429)
Special Considerations (see page 438)
Virtual Resource Management Database Extracts (see page 440)
VRM Node and Access Transport Program (see page 445)
Automated Recovery Facility Database Extracts (see page 449)

Overview

Important! CA Workload Automation SE r11.1 provided cross-platform job definition through the XPJOB command. In CA Workload Automation SE r11.3, the new AGJOB command further enhances cross-platform job definition. Jobs defined using the XPJOB command, or the equivalent DB.10 panel, cannot be transported to a pre-r11.1 level. Jobs defined using the AGJOB command, or the equivalent DB.11 panel, cannot be transported to a pre-r11.3 level.

Starting with CA Workload Automation SE r11, data set DSNBRs could exceed one million. As a result, some massaging of the CA Workload Automation SE commands may need to be done if transporting jobs from an r11 release to a pre-r11 release of CA Workload Automation SE. If this action must occur, scan the data being used to import the jobs to the pre-r11 release and ensure that data set DSNBRs, if coded, are of the format DS*nnnnnn* (six numbers instead of eight numbers)

Occasionally, you may want or need to transfer work from one CA Workload Automation SE database to another. One reason for doing this might be to better balance the workload across multiple data centers. Another reason might be to move test applications into a production database from a test database. Certainly, other reasons exist for transferring information between databases.

The effort required to redefine *manually* the workload into another database, even with online preformatted panels, 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 Workload Automation SE provides database transportability programs that, in batch mode and with the Batch Terminal Interface (BTI) facility, provide assistance in transfers of workload definitions.

The transportability process uses standard CA Workload Automation SE inquiry, Batch Terminal Interface (BTI) and Database Maintenance (DBM) facilities to assist the user in relocating workload definitions from one CA Workload Automation SE 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 the following and some other related items are beyond the realm of the functions performed by the database transportability process:

- Application data sets
- Program libraries

This transportability process is only intended to simplify the CA Workload Automation SE 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 that 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 the following:

- 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, [assign the PANVLT for your book], and PROCLIB members. (Movement of JCL can be suppressed if the user wants to do that external to this process.)

These command data sets can be used anytime after the user has reviewed, has altered, or both 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 that 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 that 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.

Assumptions

Not all of the process is automated. Some manual effort is required to ensure that the total problem is properly solved. Other assumptions are also made that the user must consider when using this process. The following address some of the particular considerations for database movement using this process.

LOAD/RELOAD Status

All CPU jobs must have been loaded with the current JCL to ensure that all data sets used by the jobs 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 that 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 that also have documentation and special characteristics.

CA WA Restart Option Step Insertion

Whenever a job is defined in the original site database with INSERT-RMS=Y to cause the CA WA Restart Option step to be automatically inserted by CA Workload Automation SE, the job is moved assuming that CA WA Restart Option 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.

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 or that perpetual calendars are defined. They are further assumed to have the same names 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.

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.)

DB.2.7 Schedule Modifications

This process has no facility to reapply any schedule changes that 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 that 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.

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.

For a discussion of the NXTCYC command, see the *Command Reference Guide*.

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. 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 because 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 because 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.

The DBT discussion repeatedly uses JCL references, assuming jobs are CPU jobs. However, when thinking of internal cross-platform jobs, the reader can substitute PARMLIB or the parameters in the sentence without hesitation.

More information:

Cataloged Procedures (see page 392)

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.

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 that execute procedures are assumed to be executing cataloged procedures and are handled as outlined in the prior discussion.

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 other networks are to be moved, the user must handle them manually.

More information:

Unconnected Workstation Networks (see page 408)

User Level Documentation

This process ignores any documentation that was defined through the DB.4.3 panel. If any such documentation is to be moved with the definition of the jobs, the customer must handle that manually, external to this process.

XPJOB/AGJOB Considerations

The XPJOB and AGJOB function DD is the same as Delete. In database transportability (DBT), when deleting either job type from the source database, the Delete function is always used, regardless of the parameter JOBDD=Y. Thus, you may see the function DD used for a CPU job with JOBDD=Y, while seeing the function Delete with an XPJOB or AGJOB job.

ANALYZE Application

To ensure database accuracy, the TRIG and RQMT functions of the CA Workload Automation SE 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 that 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 that uses them.

Commas in the Data

Because 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.

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.

User ID Security

All CPU jobs moved by this process carry the same User ID value to the new database. (For more details on this facility, see the UID field on the DB.1 panel.) 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.

More information:

DB.1 - CPU Job Definition Panel (see page 41)

LINKed Documentation

Any documentation linked using the LINK field on any of the DB.4 panels cannot be linked by this process because 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 that 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.

Database Extraction

Database extraction is an iterative process. It uses multiple BTI runs alternated with steps that examine database data. These steps produce command data sets that 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.

You provide standard LJOB commands, LSYS commands, or both to first identify those jobs that are to be moved. These commands can use any combination of selection options but *must* omit the LIST= parameter because later programs expect the default formats.

Note: For more information about these commands, see the *Command Reference Guide*.

A BTI run is made using those commands. The program SASSDT10 that produces a control report titled "Jobs Requested to be Moved" examines output from those commands. The following is an example of this report:

SASSDT10-01 Jobs Requested to be Moved

```
SASSDT10-01          CA-7 DATA BASE 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
  XPSICOM1 000 ........          000 XPJ  *NONE* *NONE* 000 000 0000  yyddd/1724

  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 jobs 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. LIST=XPJOB replaces LIST=STEPDD for XPJOBs. LIST=AGENT replaces LIST=STEPDD for AGJOBs. It is this examination of the triggers and requirements that makes it possible for the program to automatically include any workstation networks that 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 commands, LSYS commands, or both 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 that 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 that listed the component to be moved. If no commands were generated for any of the incoming commands, the characters **NONE** appear.

SASSDT20-01 Expansion of Requested Jobs

```
SASSDT20-01          CA-7 DATA BASE TRANSPORTABILITY           PAGE NO.   29
DATE: mm/dd/yy             EXPANSION OF REQUESTED JOBS

JOBNAME
     *- GENERATED COMMANDS ------------------------------------------* 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=DS00008162,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.

Two data sets contain add type commands for the new CA Workload Automation SE database. That is necessary because the processing sequence of the commands is not the same as the sequence in which the data becomes available to the programs that 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.

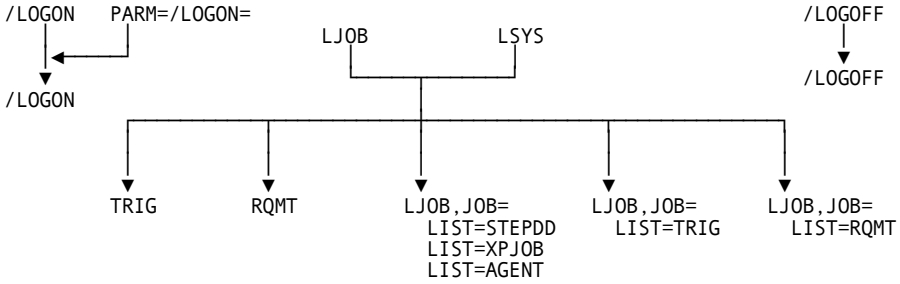
These data sets can be used whenever and wherever they are needed to accomplish the appropriate database activity. They probably require several changes that the user must make manually, using some text editor. Each data set created has a corresponding control report to allow the user to review the data 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 Workload Automation SE terminal to verify this process.

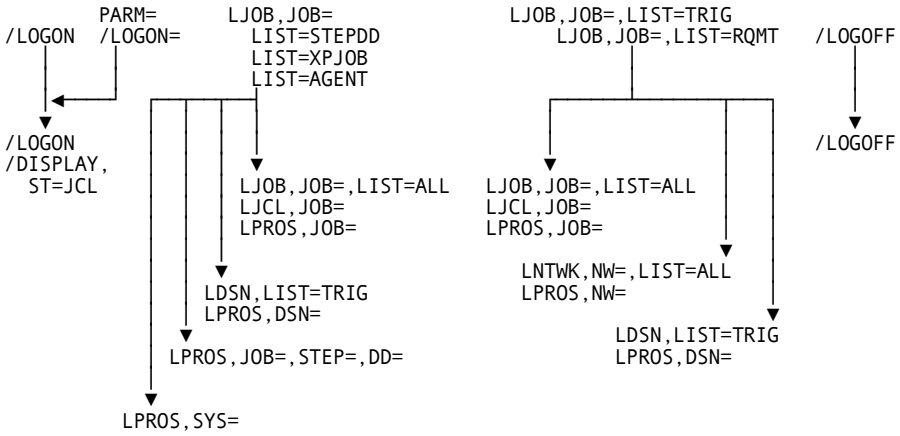
Command Evolution Process

The following topics discuss the specific types of workload definitions this process handles and describe what is produced to permit moving the workload definitions.

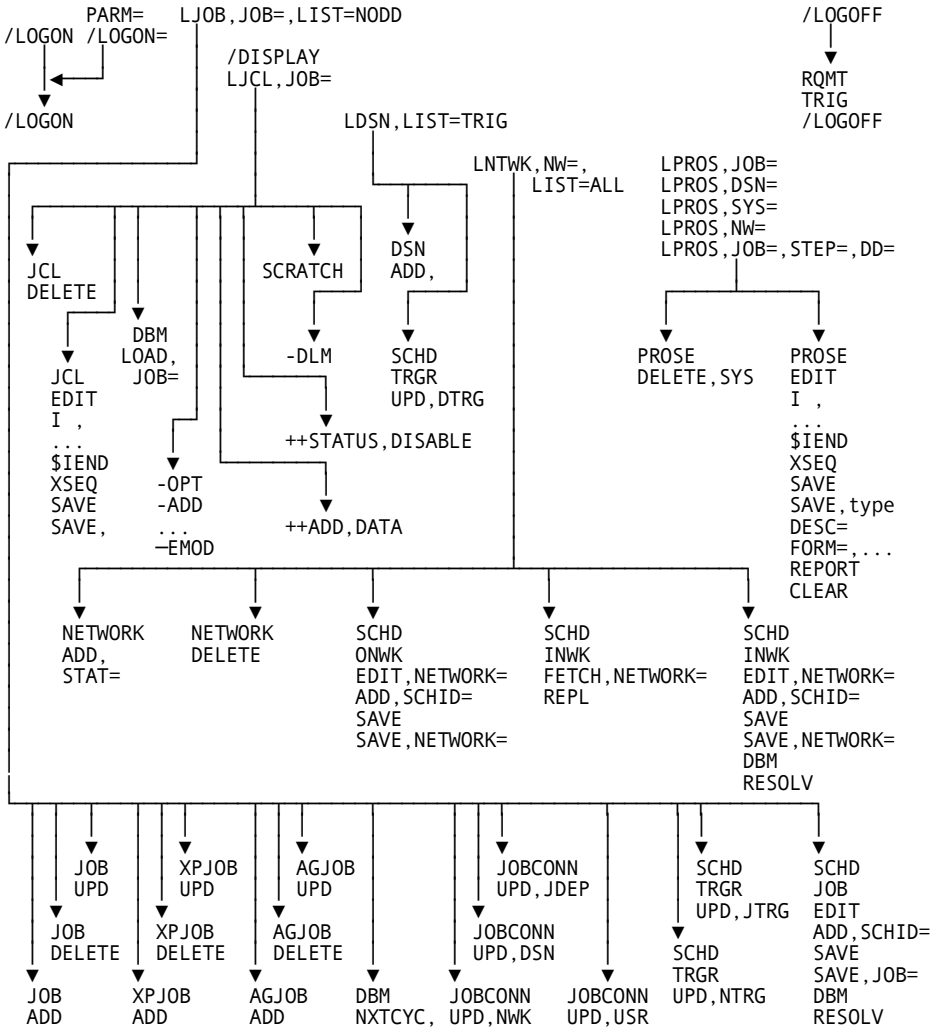
Module SASSDT10:



Module SASSDT20:



Module SASSDT30:



Moving Jobs

The following topics discuss adding jobs at the new site, deleting jobs at the original site, and disabling jobs at the original site.

Add at New Site

For each 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:

For CPU jobs:

```
JOB
ADD,jobname,.........keywords and values..............,
.
...............more keywords and values..............,
.
```

For XPJOB jobs:

```
XPJOB
ADD,jobname,.........keywords and values..............,
.
...............more keywords and values..............,
.
```

For agent jobs:

```
AGJOB
ADD,jobname,.........keywords and values..............,
.
...............more keywords and values..............,
.
```

Some historical information such as number of times run and number of times late are not carried forward to the new site because the DB.1 panel has no input fields for those values. The average CPU and elapsed time requirements are carried forward.

A number of keywords 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 Workload Automation SE 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. The RELOAD option is bypassed for both XPJOB and AGJOB jobs because they have no real JCL to load.

Because it is likely that the JCL library ID values are different at the new site, ADD commands for all 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:

SASSDT30-01 Data Base Adds - Part 1

```
SASSDT30-01          CA-7 DATA BASE TRANSPORTABILITY          PAGE NO.  265
DATE: mm/dd/yy             DATA BASE ADDS - PART 1

*- COMMANDS GENERATED ------------------------------------------------* 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
 XPJOB                                                                  U7017356
 ADD,XPSICOM1,CLASS=A,LTERM=MASTER,                                     U7017357
 NODE=TESTPLN,                                                          U7017358
 EX1=(c:\nsm\bin\cau9test                                          ),   U7017359
 XPTRACE=N                                                              U7017360
 AGJOB                                                                  U7017361
 ADD,CA750006,MEMBER=CA750006,PRMLIB=200,CLASS=A,LTERM=MASTER,          U7017362
 AGJOBTYP=UNIX_JOB,                                                     U7017363
 AGENT=UNIXAGENT,                                                       U7017364
 AG1=(USERID0006                                        )               U7017365


  NO. OF 'ADDS' FOR.
              JOBS-    175     SYS PROSE-      2
               JCL-    136     JOB PROSE-    174
              NWKS-    248     NWK PROSE-    248
              DSNS-    421      DD PROSE-      0
                               DSN PROSE-    329
```

More information:

DB.1 - CPU Job Definition Panel (see page 41)
DB.10 - XP Job Definition Panel (see page 57)
DB.11 - Agent Job Definition Panel (see page 71)

Requirements and Network Connections

Each 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 Workload Automation SE. 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.

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.

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=xxxxxxxx]  [,DESC=description]
```

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]
```

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:

SASSDT30-02 Data Base Adds - Part 2

```
SASSDT30-02          CA-7 DATA BASE TRANSPORTABILITY          PAGE NO.   119
DATE: mm/dd/yy             DATA BASE ADDS - PART 2

*- COMMANDS GENERATED ------------------------------------------------* 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/JOBS-     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
```

Delete 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 or XPJOB deletion function at the original site:

```
JOB
DELETE,jobname

XPJOB
DELETE,jobname

AGJOB
DELETE,jobname
```

The commands are sequence numbered in positions 73 through 80 and listed on the SASSDT30-03 Database Deletes report. Some other deletion functions are also listed on that report because 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 panel 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 wants to use that function instead of the default DELETE function, the JOBDD keyword must be specified in the PARM data for Job 3.

SASSDT30-03 Data Base Deletes

```
SASSDT30-03          CA-7 DATA BASE TRANSPORTABILITY          PAGE NO.    18
DATE: mm/dd/yy                DATA BASE DELETES

*- COMMANDS GENERATED ------------------------------------------------- SEQ.-NO.

JOB                                                                     U7001119
DELETE,DUSAXX09                                                         U7001120
JCL                                                                     U7001121
DELETE,DUSAXX09,DSN=PAYROLL.JCLLIB,VOL=M80007                           U7001122
XPJOB                                                                   U7001123
DELETE,XPSICOM1                                                         U7001124
AGJOB                                                                   U7001125
DELETE,AGJOB01                                                          U7001126

 NO. OF 'DELETES' FOR:
                JOBS-    175
                 JCL-    136
                NWKS-    248
           SYS/PROSE-      2
```

More information:

Job 3 (see page 433)

Disable 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 Workload Automation SE after the work has been moved.

These commands are generated as follows:

```
JOB
UPD,jobname,ADATE=yyddd,ATIME=hhmm

XPJOB
UPD,jobname,ADATE=yyddd,ATIME=hhmm

AGJOB
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.

Sequence numbers are placed in positions 73 through 80 and the commands are listed on the SASSDT30-04 control report. The following is an example of that report:

(Any commands generated to disable workstation network schedules are also listed on that report because they are also generated into the DBMSTOPS data set.)

SASSDT30-04 Data Base Disables

```
SASSDT30-04          CA-7 DATA BASE TRANSPORTABILITY          PAGE NO.    22
DATE: mm/dd/yy                DATA BASE DISABLES

*- COMMANDS GENERATED ------------------------------------------------* SEQ.-NO.

SCHD                                                                   U7001098
INWK                                                                   U7001099
FETCH,NETWORK=TIMECARD                                                 U7001100
REPL                                                                   U7001101
JOB                                                                    U7001102
UPD,DUSAXX09,ADATE=09027,ATIME=1452                                    U7001103
XPJOB                                                                  U7001104
UPD,XPSICOM1,ADATE=09027,ATIME=1452                                    U7001105
AGJOB                                                                  U7001106
UPD,AGJOB01,ADATE=09027,ATIME=1452                                     U7001107


 NO. OF 'DISABLES' FOR:
                 JOBS-    175
           SCHD/INWKS-    248
```

More information:

Job 3 (see page 433)

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.

More information:

Unconnected Workstation Networks (see page 408)

Add 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)
```

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.

Note: For more information about the LNTWK command, see the *Command Reference Guide*.

Each of these commands is sequence numbered in positions 73 through 80 and each is listed on the SASSDT30-01 control report.

More information:

Network Adds (see page 291)
DB.5 - Input/Output Network Definition Panel (see page 292)

Delete 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.

Disable at Original Site

The following topics discuss disabling networks at the original site.

Input Workstation Networks

Workstation networks have no "don't schedule after" facility. 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.

Output Workstation Networks

Because 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.

Unconnected Workstation Networks

To include any workstation networks that cannot be automatically included based on a connection or trigger definition, add two commands to the input (SYSIN DD) to Job 3, 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.

More information:

Create Command Files (see page 429)
Command Data Sets (see page 435)

Application Data Sets

The following topics address application data sets.

Add at New Site

Because all of the CPU jobs moved to the new site go through the CA Workload Automation SE 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 Workload Automation SE globally
- Data set is permanent to individual specific jobs
- Data set has documentation defined for it
- Data set triggers jobs

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
```

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, any schedule triggering can be defined, or both. Each data set is automatically assigned a new CA Workload Automation SE data set number when the definition is made at the new site.

The TYPE=PERM parameter is only included for those data sets that were so designated in the old database.

The commands are sequence numbered in positions 73 through 80. They are also listed on the SASSDT30-01 control report.

More information:

DB.6 - Data Set Definition Panel (see page 299)
Data Set Adds (see page 297)

Delete at Original Site

No commands are generated to delete data set definitions at the original site. The DB.1 panel 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, consider the following:

- Use the JOBDD keyword in the third job of the generation process, or
- Edit manually the generated commands to delete *jobs* from the database at the original site to replace the function value DELETE with DD before using the generated commands.

Job 2 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 that this process is moving.

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.

More information:

Job 2 (see page 432)

PDS JCL Members

If the user wants 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.

The discussions in this section assume that the user wants to produce reports and commands.

More information:

Create Command Files (see page 429)

Add at New Site

If a 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.

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. 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.

More information:

JCL and PARM Data Maintenance (see page 305)

Delete 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 Workload Automation SE.

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.

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.

CA Librarian JCL

If the user wants 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.

The discussions in this section assume that the user wants to produce reports and commands.

More information:

Create Command Files (see page 429)

Add at New Site

If a 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. For the meanings of the various commands and parameters used here, see the CA Librarian publications. -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 that is produced on the file referenced by the ddname DT30CR06. The actual JCL statements and the -EMOD statements are *not* listed.

Any other CA Librarian control statements can be added to this data set prior to using it at the new site. Because CA Workload Automation SE 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. To determine how this is done, see the CA Librarian publications.

The new library at the new site and the JCLID or JCLLIB values in the JOB commands generated for the jobs that 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 Librarian JCL Member Adds

```
SASSDT30-06          CA-7 DATA BASE TRANSPORTABILITY          PAGE NO.   48
DATE: mm/dd/yy            LIBRARIAN JCL MEMBER ADDS

*- COMMANDS GENERATED ------------------------------------------------* 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
```

Delete 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. For other requirements, see the CA Librarian publications.

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 Librarian JCL Member Deletes

```
SASSDT30-07          CA-7 DATA BASE TRANSPORTABILITY          PAGE NO.   24
DATE: mm/dd/yy           LIBRARIAN JCL MEMBER DELETES

*- COMMANDS GENERATED ----------------------------------------------* 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
```

CA Panvalet JCL

If the user wants 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.

The discussions in this section assume that the user wants to produce reports and commands.

More information:

Create Command Files (see page 429)

Add at New Site

If a job has an execution JCL member that resides in a [assign the PANVLT for your book] 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 that is produced on the file referenced by the ddname of DT30CR08. The JCL statements themselves are *not* listed.

Any other [assign the PANVLT for your book] control statements that the user may require can be added to this data set before using it at the new site. Because CA Workload Automation SE does not provide update access to [assign the PANVLT for your book] libraries, the user *must* assemble whatever job is needed to add this JCL to another [assign the PANVLT for your book] library at the new site. To determine how this is done, see the [assign the PANVLT for your book] publications.

The new library at the new site and the JCLID values in the JOB commands generated for the jobs that 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 Panvalet JCL Member Adds

```
SASSDT30-08          CA-7 DATA BASE TRANSPORTABILITY          PAGE NO.   24
DATE: mm/dd/yy             PANVALET JCL MEMBER ADDS

*- COMMANDS GENERATED ------------------------------------------------* 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
```

Disable 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. For the meaning of the DISABLE parameter and any other requirements that exist for this purpose, see the [assign the PANVLT for your book] publications.

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 Panvalet JCL Member Deletes

```
SASSDT30-09          CA-7 DATA BASE TRANSPORTABILITY          PAGE NO.   24
DATE: mm/dd/yy           PANVALET JCL MEMBER DELETES

*- COMMANDS GENERATED ------------------------------------------------* 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
```

Cataloged Procedures

If the user wants 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.

The discussions in this section assume that the user wants to produce reports and commands.

More information:

Create Command Files (see page 429)

Add at New Site

Normally, PROCLIBs are not made available to CA Workload Automation SE because 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.

Delete at Original Site

Any PROCs executed by the JCL for any jobs being moved are candidates for deletion at the original site. Because 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.

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 that referenced the PROC. The character string vvvvvv and space for a data set name value are provided in the statements. (CA Workload Automation SE 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 DATA BASE TRANSPORTABILITY          PAGE NO.   24
DATE: mm/dd/yy              PROCLIB MEMBER DELETES

*- COMMANDS GENERATED ------------------------------------------------* 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
```

Documentation

The following topics address documentation.

Add at New Site

CA Workload Automation SE supports six categories of user-defined documentation. The transportability process handles five of these. Any user level documentation (as defined through the DB.4.3 panel) 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. Any documentation extending past position 72 is truncated when moved to the new CA Workload Automation SE database.

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

For a user option for DD level documentation movement, see the DDPROSE keyword in Job 2.

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 commands 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. The other commands are only written to the DBMADDS1 data set.

More information:

Job 2 (see page 432)

Delete 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.

Schedules

The following topics address schedules.

Add at New Site

The following topics address changes related to job schedules, job triggers, network triggers, data set triggers, input networks, and output networks.

Job Schedules

For defined schedule IDs for those 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
```

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 multiple schedule IDs are 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.

If the job had a current schedule modification, made with the online DB.2.7 panel 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.

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]
```

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.

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]
```

The commands are sequence numbered in positions 73 through 80 and are listed on the SASSDT30-02 control report.

Data Set Triggers

For each 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]
```

Because 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.

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=yy,ROLL=x,INDEX=>nnn
DBM
RESOLV,SCAL=yy,TEST=NO,NW=NW.networkname
```

Also, a DB.2.7 warning message sometimes appears on the control report, which we discussed previously.

The ADD commands are repeated as many times as necessary between the EDIT and SAVE commands to accommodate all schedule IDs to be defined. All parameter values used are as they were defined in the original database *including* the Base Calendar ID that 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.

Each command is sequence numbered in positions 73 through 80 and is listed on the SASSDT30-02 control report.

More information:

Job Schedules (see page 423)

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.

Delete at Original Site

The following topics discuss deleting at the original site.

Job Schedules

Because the schedules for jobs are automatically deleted whenever the job definition is deleted from the database, this process generates no additional commands to delete job schedules.

More information:

Disable at Original Site (see page 405)

Input Network Schedules

Because 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.

Output Network Schedules

Because output networks are scheduled whenever their associated CPU job is scheduled, no extra commands are necessary.

More information:

Job Schedules (see page 423)

Node and Access File

Each XPJOB job that is selected for transport has its OWNER and XP NODE information written to the VRMXREFS data set. The output has the following format:

N*xxxxx*

Identifies the node name.

ON*xxxxx*

Identifies the node name.

OO*xxxxx*

Identifies the owner.

The N, ON, and OO prefixes indicate the record type in the database. These records are used during the VRM database extract to generate transport transactions for Node records (created using the XNODE command) and access records (created using the XPSWD command). This is discussed in detail later in this section.

Each cross reference record is listed on the SASSDT30-10 control report produced on the file referenced by the ddname DT30CR10. Following is a sample of the SASSDT30-10 report.

SASSDT30-10 VRM Cross Reference Entries

```
SASSDT30-10          CA-7 DATA  BASE TRANSPORTABILITY          PAGE NO.     1
DATE:  8/11/yy           VRM CROSS REFERENCE ENTRIES
*- COMMANDS GENERATED ------------------------------------------------* SEQ.-NO
NTESTPLN                                                               U7000001
ONTESTPLN                                                              U7000002
OOCHRIS                                                                U7000003
NBNODE2                                                                U7000004
ONBNODE2                                                               U7000005
OOPAT                                                                  U7000006
NCNODE1                                                                U7000007
ONCNODE1                                                               U7000008
OOSAM                                                                  U7000009
```

Agent DIV File

AGJOB jobs that are selected for transport may also have selected information written to the AGENTDIV data set. This only occurs if the USER field in the Agent job is not blank. In these situations, the Jobname, Agent Job Type, Agent, and User (as seen on the DB.11 panel) are written to the AGENTDIV data set as a single line.

This file lets your site's security administrator cross reference this information with security records that may have been added using the AGPSWD command. Agent jobs typically require a User ID and Password to run on the remote platform. CA Workload Automation SE uses the AGPSWD command to define these entities. Site personnel who have authority to issue the AGPSWD command must manually set up these security definitions on the CA Workload Automation SE instance to which agent jobs are being transported. We recommend using the AGENTDIV file output to assist in determining what security definitions are required.

In addition, for each record written to the AGENTDIV file, three lines are written to the SASSDT30-11 control report that is referenced by ddname DT30CR11:

- The first line contains the job name, agent job type, and agent.
- The second line contains the first 64 characters of the User field.
- The third line contains the last 64 characters of the User field or blanks.

Following is a sample of the SASSDT30-11 report.

SASSDT30-11 Agent Jobs DIV File XREF

```
SASSDT30-11          CA-7 DATA  BASE TRANSPORTABILITY           PAGE NO.     1
DATE: mm/dd/yy           AGENT JOBS DIV FILE XREF

*- COMMANDS GENERATED ------------------------------------------------* SEQ.-NO.
CA750003  UNIX_JOB         UNIXAGENT                                     U7000001
userid000300000000000000000000000000000000000000000000000000000003       U7000002
000000000000000000000000000000000000000000000000000000000000000003       U7000003
CA750006  UNIX_JOB         UNIXAGENT                                     U7000004
userid000600000000000000000000000000000000000000000000000000000006       U7000005
                                                                          U7000006
```

Create Command Files

The transportability process consists of multiple steps. It could be done with a single job. The user probably wants to develop multiple jobs, however, so that the processing can be manually verified between each of the three major iterations against the database.

During CA Workload Automation SE installation, the CA Workload Automation SE SYSGEN created three jobs: CA07N810, CA07N820, and CA07N830, which can be used for CA Workload Automation SE transportability processing.

The following discussions and examples assume that the user uses three different jobs to perform this process. JCL parameters that 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.

Note: For more information about this initialization file parameter, see the *Systems Programming Guide*.

More information:

Special Considerations (see page 438)

Job 1

Refer to the CA07N810 JCLLIB member (from the SYSGEN process) for a sample JCL member. The job has two steps.

The first step of this job is a standard BTI step.

Note: For more information about this step, see "Using Batch Terminals" in the *Interface Reference Guide*.

PARM Keywords

A CA Workload Automation SE 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 Workload Automation SE, including the comma that separates the operator ID and any password value being provided. Operator ID can be up to eight characters in length. A maximum total of 65 characters is allowed for the operator ID, the password value and the comma that 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).

Specifying Jobs to Move

The input to the BTI run is in the form of CA Workload Automation SE LJOB and/or LSYS commands. They may be used in any combination to define the jobs to be moved. If any workstation networks are to also be moved, the programs automatically include those with a predecessor or successor relationship with any of the jobs specified. If others are not connected to a job, they must be moved manually by the user.

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 jobs 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 because 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 that 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 that 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.

More information:

Workstation Networks (see page 406)

Data Sets Used/Created

The SYSPRINT output is being passed to the second step as that step's input.

The second 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.

The following is a SASSDT10 execution JCL sample:

```
//jobname  JOB .........,REGION=1024K
//EXTRACT  EXEC PGM=SASSBSTR,PARM=batch-terminal-id
//JOBLIB   DD  DSN=cai.CAL2LOAD,DISP=SHR
//UCC7CMDS DD  DSN=ca7.communications.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
/*
//CREATE1  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
```

Job 2

After you verify that the jobs were identified correctly in Job 1, you can run Job 2.

Refer to the CA07N820 JCLLIB member (from the SYSGEN process) for a sample JCL member. The job also has two steps.

This job performs another iteration against the database much like the first job. The commands in the SYSIN data set for SASSBSTR produce listings for JCL, trigger, and requirement information for each of the jobs to move. Module SASSDT20 examines the BTI output (ddname DT20IN) and produces another card-image data set to be used in Job 3 (ddname DT20OUT).

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 Job 1.

The user also has an option available that 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 panel, 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. (Because 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 ignore totally the movement of JCL, XPJOB PARM data, or AGJOB PARM data. Code the parameter JCL=N in the PARM data for module SASSDT20 to accomplish this. N is the only acceptable value. With this option specified, JCL and PARM data are ignored here, and the user must handle the movement of these elements totally external to this process. JCL=N in this job suppresses all of the JCL and PARM data 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.

More information:

Job 1 (see page 429)

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.

After verifying that Job 2 ran correctly, Job 3 may be run.

The following is a SASSDT20 execution JCL sample:

```
//jobname  JOB .......,REGION=4M
//EXTRACT  EXEC PGM=SASSBSTR,PARM=batch-terminal-id
//JOBLIB   DD  DSN=cai.CAL2LOAD,DISP=SHR
//UCC7CMDS DD  DSN=ca7.communications.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)
//*
//CREATE2  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
```

Job 3

Refer to the CA07N830 JCLLIB member (from the SYSGEN process) for a sample JCL member. The job has two steps.

This job performs another iteration against the database, much like the first two jobs. The commands in the SYSIN data set, from Job 2, produce listings for other database elements that are to move. The user can 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.

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 Job 1.

The user also has the following optional keywords that can be provided in the PARM for the purposes indicated:

JOBDD=Y

Specifies 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.

Note: For more information about this function, see the job definition panels.

Y is the only acceptable value.

AFTER=*yyddd*[*hhmm*]

Specifies "don't schedule after" values to use for the disable commands generated to disable 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).

Note: For more information about this function, see the job definition panels.

BEFORE=*yyddd*[*hhmm*]

Specifies "don't schedule before" values to use for the add commands generated to add 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).

Note: For more information about this function, see the job definition panels.

LOADS=Y

Specifies 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 because 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

Specifies that DSN ADD commands are not generated in the output of this run. This parameter can be used if data set definitions are not required at the receiving site. By default, data set ADDs are generated.

DFLTOVRD=Y

Generates explicit keyword/value pairs (with default values) for most job definition panel fields (see the following Exclude List).

Typically, explicit keyword/value pairs for job definition panel fields (and their values) are *not* generated if the value specified matches the default value.

Generation of explicit keyword/value pairs is useful when the receiving CA Workload Automation SE system's DEFAULTJOB/DEFAULTXPJ/DEFAULTAG (see the DBASE initialization statement) settings are different from the sending system's settings, and the original job definition settings are still wanted. Y is the only acceptable value for this parameter.

Exclude List: LIB, USE-OVR-LIB, SATISFACTION LEAD TIME (JOB & DSN), COND-CODE, RO, DON'T SCHEDULE BEFORE/AFTER, LTERM, MSGCLASS.

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
DBMADDS1	DBM add commands - Part 1	DT30CR01
DBMADDS2	DBM add commands - Part 2	DT30CR02
DBMDELTS	DBM delete commands	DT30CR03
DBMSTOPS	DBM disable commands	DT30CR04
DELPROCS	SCRATCH commands for PROCs	DT30CR05
LIBRADDS	CA Librarian JCL add modules	DT30CR06
LIBRDELS	CA Librarian delete commands	DT30CR07
PANVADDS	[assign the PANVLT for your book] JCL add modules	DT30CR08
PANVDELS	[assign the PANVLT for your book]disable commands	DT30CR09
VRMXREFS	VRM cross-reference entries	DT30CR10
AGENTDIV	Agent Jobs DIV File XREF	DT30CR11

Details on the contents of these data sets and the examples of these control reports were given previously.

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. 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 [assign the PANVLT for your book] or CA Librarian JCL is being used, those data sets and their control reports may be DUMMYed or defined as NULLFILEs.

The following is a SASSDT30 execution JCL sample:

```
//jobname  JOB ......,REGION=4096K
//EXTRACT  EXEC PGM=SASSBSTR,REGION=1024K,PARM=batch-terminal-id
//JOBLIB   DD  DSN=cai.CAL2LOAD,DISP=SHR
//UCC7CMDS DD  DSN=ca7.communications.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)
//*
//CREATE3  EXEC PGM=SASSDT30,COND=(0,NE,EXTRACT),
//             PARM='LOGON=operid[,password].............'
//SYSUDUMP DD  SYSOUT=x
//DBMLISTS DD  DSN=&&DT30IN,DISP=(OLD,DELETE)
//DBMADDS1 DD  DSN=batch.DBM.add.commands.part.1,DISP=(,CATLG),UNIT=SYSDA,
//             SPACE=(CYL,(5,1),RLSE),DCB=(RECFM=FB,LRECL=80,BLKSIZE=4000)
//DBMADDS2 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)
//LIBRADDS DD  DSN=librarian.add.commands,DISP=(,CATLG),UNIT=SYSDA,
//             SPACE=(CYL,(1,1),RLSE),DCB=(RECFM=FB,LRECL=80,BLKSIZE=4000)
//LIBRDELS DD  DSN=librarian.delete.commands,DISP=(,CATLG),UNIT=SYSDA,
//             SPACE=(CYL,(1,1),RLSE),DCB=(RECFM=FB,LRECL=80,BLKSIZE=4000)
//PANVADDS DD  DSN=panvalet.add.commands,DISP=(,CATLG),UNIT=SYSDA,
//             SPACE=(CYL,(1,1),RLSE),DCB=(RECFM=FB,LRECL=80,BLKSIZE=4000)
//PANVDELS DD  DSN=panvalet.disable.commands,DISP=(,CATLG),UNIT=SYSDA,
//             SPACE=(CYL,(1,1),RLSE),DCB=(RECFM=FB,LRECL=80,BLKSIZE=4000)
//VRMXREFS DD  DSN=vrm.cross.reference.list,DISP=(,CATLG),UNIT=SYSDA,
//             SPACE=(CYL,(1,1),RLSE),DCB=(RECFM=FB,LRECL=80,BLKSIZE=4000)
//AGENTDIV DD  DSN=agentdiv.cross.reference.list,DISP=(,CATLG),UNIT=SYSDA,
//             SPACE=(CYL,(1,1),RLSE),DCB=(RECFM=FB,LRECL=172,BLKSIZE=0)
//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
//DT30CR10 DD  SYSOUT=x,DCB=BLKSIZE=133
//DT30CR11 DD  SYSOUT=x,DCB=BLKSIZE=133
```

More information:

Assumptions (see page 389)
Moving Jobs (see page 399)
Workstation Networks (see page 406)
Run at New Site (see page 440)

Special Considerations

Keep the following special considerations in mind during this process.

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 that constitute this process uses tables for these purposes as follows:

Module	Maximum#	Component
SASSDT10	50,000	Job names
SASSDT20	5,000	Application system names
SASSDT20	50,000	Job names
SASSDT20	10,000	Workstation network names
SASSDT20	300,000	CA Workload Automation SE 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.

Add 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 Workload Automation SE at that site. Any JCL libraries for [assign the PANVLT for your book] or CA Librarian JCL also must be processed at that site. Either a batch terminal job or a batch execution of CA Workload Automation SE could be used to accomplish this. If using the batch execution of CA Workload Automation SE, see the *Systems Programming Guide* 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. For guidelines about calculating space requirements in the database, see the *Systems Programming Guide*.

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. Because documentation is added one record at a time, with a response back after each line in batch, this process 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. Users must resolve any duplication of names for jobs, networks, and other items manually.

Run 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 jobs can include the "don't schedule before" parameters, BDATE and BTIME, to accomplish this for 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.

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 sections, 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.

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. It also selects node and access resources that match node and access records extracted by the SASSDT30 program. Three files, each containing a set of CA Workload Automation SE Batch Terminal Interface commands in a fixed-block file format, are generated for transporting information to a different database.

During CA Workload Automation SE installation, the CA Workload Automation SE SYSGEN created three jobs: CA07N840, CA07N841, and CA07N845, which can be used for CA Workload Automation SE VRM database transportability processing.

VRM DBT Program

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 that contains the CA Workload Automation SE Batch Terminal Interface commands required to add the job to resource connections to a different VRM database component. In addition to the CA Workload Automation SE Batch Terminal Interface command file, a job selection report SASSDT60-01 is generated to report on any or all jobs selected for this run.

The following is a SASSDT60-01 VRM Job Extract report sample:

```
SASSDT60-01                                  CA-7 DATA BASE TRANSPORTABILITY                          PAGE NO. 00001
DATE: 10/22/yy                                   VRM JOB EXTRACT REPORT


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
             PAYROLL.FILES.PAYCHECK                 200       NODATA        EXC      ABNORMAL TERMINATION
PAYROLL2     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
             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
PAYROLL3     TL6642.TEST.DATA.PAY3                  238       JS20          EXC      SUCCESSFUL COMPLETION OF JOB STEP
             SYSX.PROCS.USER.DATA                   000       JS30          SHR      ALWAYS FREE RESOURCE
             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
PAYROLL4     HELP.TEMP.DATASET1                     000       JS10          EXC      SUCCESSFUL COMPLETION OF JOB STEP
             HELP.TEMP.DATASET2                     000       JS20          EXC      SUCCESSFUL COMPLETION OF JOB STEP
             SYS1.DUMPIT.DATA.TEMP.DATA             000                     SHR      ABNORMAL TERMINATION
             SYSX.TL6642.HELP                       000       JS30          SHR      FREE RESOURCE MANUALLY
             SYS2.SYS2.SYS2.SYS2.SYS2               000                     EXC      ABNORMAL TERMINATION
PAYROLL6     CICSREG1                               000       STEP0099      ASX      FREE RESOURCE MANUALLY
PAYROLL7     CICSREG8                               000                     ASX      RESOURCE MUST BE INACTIVE
             DB2                                    000                     CRQ      RESOURCE MUST BE INACTIVE


TOTAL JOB RECORDS EXTRACTED .... 00006
```

VRM DBT Job Extract JCL

The first step in Virtual Resource Management Database Transportability is execution of the SASSDT60 program.

The following is a SASSDT60 execution JCL sample:

```
//jobname  JOB .........
//VRMDBT  EXEC PGM=SASSDT60,REGION=1024K,
//    PARM='ALLJOBS'
//STEPLIB   DD DISP=SHR,
//             DSN=cai.CAL2LOAD
//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
/*
//
```

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 Workload Automation SE 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.

The following are PARM examples:

```
//jobname job .........
//  EXEC PGM=SASSDT60,PARM='REPORT'
```

```
//jobname job .........
//  EXEC PGM=SASSDT60,PARM='REPORT,ALLJOBS'
```

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 characters. 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 has no limit.

The following are SYSIN DD control card examples:

```
//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*
```

Data Sets Used/Created

CA7RSRC

This is the CA Workload Automation SE VRM database component.

DT60ADDS

This is the generated CA Workload Automation SE Batch Terminal Interface commands file. This file is used as input into a CA Workload Automation SE 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 Workload Automation SE 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 panel.

VRM Node and Access Transport Program

The SASSDT62 program is used to select node and access records that match the node and access records extracted by the SASSDT30 program. The program creates two files. The DT62NODE file contains the CA Workload Automation SE Batch Terminal Interface commands required to add node resources to a different VRM database. The DT62XPWD file contains the commands required to add the access resources. Examples of both files are shown in the following:

```
XNODE
ADD,XP63N01,
ALT1=AL6301,
ALT2=AL6301,
DES=TEST DESCRIPTION FOR NODE6301,
STATE=ONLINE

XPSWD
ADD,
NODE=XP63N01,
XPUSER=XP63U01,XPDMN=XP63D01
ADD,
OWNER=XP63O01,XPUSER=XP63U01,XPDMN=XP63D01
```

Note: STATE is always set to ONLINE, regardless of the setting in VRM.

VRM Node and Access Extract JCL

The following is a SASSDT62 execution JCL sample:

```
//*-------------------------------------------------------------------*
//*                                                                   *
//*      'SAMPLE DATABASE TRANSPORTABILITY JOB'
//*                                                                   *
//*-------------------------------------------------------------------*
//SORTIT  EXEC PGM=SORT,REGION=2048K
//SYSOUT    DD SYSOUT=A
//SYSPRINT  DD SYSOUT=A
//SORTIN    DD DSN=ca7.VRMXREFS,DISP=SHR
//SORTOUT   DD DSN=&&VRMXREFS,
//             DISP=(NEW,PASS),
//             SPACE=(TRK,(5,5)),UNIT=SYSDA,
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=0)
//SORTWK01  DD UNIT=VIO,SPACE=(CYL,(10,10))
//SORTWK02  DD UNIT=VIO,SPACE=(CYL,(10,10))
//SORTWK03  DD UNIT=VIO,SPACE=(CYL,(10,10))
//SYSIN     DD *
  SORT FIELDS=(1,66,CH,A)
//VRMDBT2  EXEC PGM=SASSDT62,REGION=4096K,
//    PARM='REPORT',COND=(04,LT)
//STEPLIB  DD DISP=SHR,DSN=user-defined-CA-7-loadlib
//DT62INPT DD DSN=&&VRMXREFS,DISP=(OLD,PASS)
//CA7RSRC  DD DSN=ca7.vrm.database,DISP=SHR
//DT62NODE DD DSN=ca7.DT62NODE,
//     DISP=(NEW,CATLG,DELETE),
//     UNIT=SYSDA,
//     SPACE=(CYL,(5,2),RLSE),
//     DCB=(RECFM=FB,LRECL=80,BLKSIZE=0)
//DT62XPWD DD DSN=ca7.DT62XPWD,
//     DISP=(NEW,CATLG,DELETE),
//     UNIT=SYSDA,
//     SPACE=(CYL,(5,2),RLSE),
//     DCB=(RECFM=FB,LRECL=80,BLKSIZE=0)
//SYSUDUMP DD SYSOUT=*
//NODEREPT DD SYSOUT=*,DCB=BLKSIZE=133
//NMTCHRPT DD SYSOUT=*,DCB=BLKSIZE=133
//NODPREPT DD SYSOUT=*,DCB=BLKSIZE=133
//OWNPREPT DD SYSOUT=*,DCB=BLKSIZE=133
```

PARM Keywords

The SASSDT62 program accepts one PARM value, which is not required. A value of REPORT indicates that this is a report run only.

A value of PASSWORD indicates that encrypted password information is to be included in the BTI transactions, assuming the records have associated passwords (the password is not required when adding records with the CA Workload Automation SE XPSWD command).

If neither parameter is specified, the BTI transactions are created without password information. Thus, you must specify the PASSWORD parameter if you wish to transport associated password information.

Password information is never displayed in the reports, regardless of the PARM setting.

The following are PARM examples:

```
//jobname job
//     EXEC PGM=SASSDT62,PARM='REPORT'

//jobname job
//     EXEC PGM=SASSDT62,PARM='PASSWORD'
```

Data Sets Used/Created

The following data sets are used and created during the SASSDT62 process:

SORTIN

This file contains the cross-reference records generated during the SASSDT30 program execution and written to the VRMXREFS DD statement.

CA7RSRC

This is the CA Workload Automation SE VRM database component.

DT62NODE

This file is the generated Batch Terminal Interface commands file for the node entries. This file is used as input into a CA Workload Automation SE BTI job to add the node resource definitions on a different database.

DT62XPWD

This file is the generated CA Workload Automation SE Batch Terminal Interface commands file for the node and owner access entries. This file is used as input into a CA Workload Automation SE BTI job to add the node and owner access resource definitions on a different VRM database.

Note: The final step of the VRM Database Transportability process is to run a Batch Terminal Interface job (CA07N845). This job consists of two steps. The first step adds the database records created in SASSDT60. The second step adds node records (DT62NODE) and access records (DT62XPWD).

The Batch Terminal Interface mimics the interactive CA Workload Automation SE commands. The XNODE command (DT62NODE) is processed to add node records. The XPSWD command (DT62XPWD) is processed to add owner and node access records. Node access records contained in the DT62XPWD file *cannot* be added when a corresponding node record does not exist. Thus, the DT62NODE file must run through the Batch Terminal Interface first.

Output Reports

SASSDT62 processing can produce the following reports:

NODEREPT is the SASSDT62-01 XPJOB Entries with Matching Node Records report file.

```
SASSDT63- 1                                  CA-7 DATA BASE TRANSPORTABILITY                                  PAGE NO. 1
DATE: 08/30/yy                         XPJOB Entries with Matching Node Records

Primary Node                                                  Description
   Alternate Node 1                                              Alternate Node 2
XP63N01                                                       DESCRIPTION FOR NODE6301
   AL6301                                                        AL6301
```

NODPREPT is the SASSDT62-02 XPJOB Entries with Matching Node Access Records report file.

```
SASSDT63- 2                                  CA-7 DATA BASE TRANSPORTABILITY                                  PAGE NO. 1
DATE: 08/30/yy                       XPJOB Entries with Matching Node Access Records

Node                                    User ID                       Domain
XP63N01                                 XP63U01                       XP63D01
XP63N02                                 XP63U02                       XP63D01
```

OWNPREPT is the SASSDT62-03 XPJOB Entries with Matching Owner Access Records report file.

```
SASSDT63- 3                                    CA-7 DATA BASE TRANSPORTABILITY                                    PAGE NO. 1
DATE: 08/30/yy                        XPJOB Entries with Matching Owner Access Records

Owner    User ID                       Domain
XP63O01  XP63U01                       XP63D01
XP63O02  XP63U02                       XP63D01
XP63O03  XP63U03                       XP63D01
```

NMTCHRPT is the SASSDT62-04 XPJOB Entries with No Matching Node or Access Records report file.

```
SASSDT63- 4                                    CA-7 DATA BASE TRANSPORTABILITY                                    PAGE NO. 1
DATE: 08/30/yy                      XPJOB Entries with No Matching Node or Access Records

No matching Node record found for XP63N100
No matching Node record found for XP63N101
No matching Access record found for Node XP63N100
No matching Access record found for Node XP63N101
No matching Access record found for Owner XP63O101
No matching Access record found for Owner XP63O102
```

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 Workload Automation SE Batch Terminal Interface commands, in a fixed file format, for transport to a different database.

During CA Workload Automation SE installation, the CA Workload Automation SE SYSGEN created two jobs: CA07N850 and CA07N855, which can be used for CA Workload Automation SE ARF database transportability processing.

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 Workload Automation SE 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 following is a SASSDT72-01 ARFSET Extract report sample:

```
SASSDT72-01                                        CA-7 DATA BASE TRANSPORTABILITY                                  PAGE NO. 00001
DATE:  9/30/20yy                                        ARFSET EXTRACT REPORT
-------------------------------------------------------------------------------------------------------------------------------
ARFSET NAME   UID   RESPONDER
ARFTST01      000    MASTER
DEFINITION INDEX
     001
TYPE    SYSTEM  REL OPR    SCHID  REL OPR    RESTART COUNT  REL OPR    ENTRY MODE  REL OPR
 JC     *         EQ        000     EQ            000         GE          *         EQ
FROM DATE    FROM TIME    TO DATE    TO TIME
01011975       0001       12312074    2359
                                                          - JOB COMPLETION CHECK -
PROCEDURE    PROC REL OPR    STEP          STEP REL OPR    PROGRAM    PGM REL OPR    COND TYPE    COND REL OPR    VALUE    MASK
*               EQ         *                 EQ           *            EQ         CC              GE            0004

RESPONSE:
COMM MODE    USERID    MESSAGE TEXT
L            USER015   JC - JOB COMPLETION CHECK
RESPONSE:
COMM MODE    USERID    MESSAGE TEXT
L            USER015   JC - JOB COMPLETION CHECK
RESPONSE:
COMM MODE    USERID    MESSAGE TEXT
L            USER015   SYSTEM=&SYSTEM SCHDID=&SCHDID JES#=&JES#
FINAL DISPOSITION
NO ACTION TAKEN
CA-11  BYPASS GDG  USAGE  PROCESS  COND CODE  START STEP         END STEP
  N         N
-------------------------------------------------------------------------------------------------------------------------------
ARFSET NAME   UID   RESPONDER
ARFTST02      000    MASTER
DEFINITION INDEX
     001
TYPE    SYSTEM  REL OPR    SCHID  REL OPR    RESTART COUNT  REL OPR    ENTRY MODE  REL OPR
 JC     *         EQ        000     EQ            000         GE          *         EQ
FROM DATE    FROM TIME    TO DATE    TO TIME
01011975       0001       12312074    2359
```

ARF DBT Extract JCL

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.CAL2LOAD
//*
//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
```

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 Workload Automation SE 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 characters. 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*
```

Data Sets Used/Created

CA7ARFDB

This is the CA Workload Automation SE ARF database component.

DT70ADDS

This is the generated CA Workload Automation SE Batch Terminal Interface commands file. This file is used as input into a CA Workload Automation SE 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 Workload Automation SE Batch Terminal Interface job using the generated DT70ADDS file as SYSIN input to add the ARFSET definitions on the target ARF database.

Chapter 15: Mass Job Rename Utility

This section contains the following topics:

Overview (see page 453)
Mass Job Rename Procedure (see page 454)
Mass Job Rename Control File (RENAMES) (see page 456)
Database Rename Process (CAL2JRXD) (see page 456)
Queue Rename Process (CAL2JRXQ) (see page 460)
VRM Process (CAL2JRXV) (see page 463)
ARF Process (CAL2JRXA) (see page 467)

Overview

The Mass Job Rename Utility consists of several batch processes that can change job names in all places where they appear in the CA Workload Automation SE database and queue files. This includes all indirect references to job names such as job predecessors, job prose, ending job names in CPM Flow definitions that are defined as of VRM Corequisite resources, and so forth. These utilities can be used to rename large numbers of job names at the same time with a single run of the utility jobs to ensure that all components of CA Workload Automation SE remain synchronized.

The following processes make up the Mass Job Rename Utility:

- Database rename processor (CAL2JRXD). This processor handles elements in the main CA Workload Automation SE Job, Dataset, and IDS files. It runs against an IDCAMS or SASSBK00 sequential backup of the CA Workload Automation SE database.
- Queue rename processor (CAL2JRXQ). This processor handles elements in the CA Workload Automation SE queue files (request, ready, active, and prior run queues). It runs against a sequential backup of the queues generated by a DMPQ shutdown of CA Workload Automation SE.
- VRM rename processor (CAL2JRXV). This processor handles elements in the CA Workload Automation SE VRM file. This utility is used if you use the CA Workload Automation SE Virtual Resource Management (VRM) feature. It runs against an IDCAMS sequential backup of the CA Workload Automation SE database.
- ARF Rename Processor (CAL2JRXA). This processor handles elements in the ARF file. This utility is used if you use the CA Workload Automation SE Automatic Recovery Facility (ARF) feature. It runs against an IDCAMS sequential backup of the ARF database.

You can obtain a preview of the changes by submitting the jobs and having the output specified as DD DUMMY,DCB=(data). The preview shows all areas where the job name was found and the new name that will be used. This way, you can review all changes prior to executing the process that updates the CA Workload Automation SE databases. It is important that all four utilities, if VRM and ARF used, are executed against the current backup files concurrently or in the same time period before CA Workload Automation SE is brought back into the system. Failure to execute the Queues Rename utility may cause missing requirements for future jobs once CA Workload Automation SE is brought back into the system.

The SUMMARY DD statement is optional for each of the utilities. If the SUMMARY DD statement is not coded, the summary report is included in the SYSPRINT DD output. The data is separated should the installation want to direct the report summary to a different location.

For specific details about the elements that can be changed through this process, see the subsequent sections in this chapter.

Note: The Mass Job Rename utilities do not examine any JCL libraries. As a result, job names that are used in the Batch Terminal Interface (BTI) programs, such as SASSTRLR and U7SVC, are not changed. If you have BTI steps in your production JCL that reference a job with a new name, you must change the production JCL to reflect the new job name.

Mass Job Rename Procedure

Each of the CA Workload Automation SE Mass Job Rename processes uses sequential backups of their respective databases as input. A user-defined control file describing the old and new job names is also supplied as input. The output is a sequential file with altered job names that is suitable for reload.

To perform a mass job rename

1. Create a RENAMES control file defining the job names to be changed with their corresponding new names. The same RENAMES control file will be used for all rename processors to ensure that the changes are consistent across all elements.
2. Shut down CA Workload Automation SE with a DMPQ option to create a backup of the CA Workload Automation SE queues.
3. Create a backup file of the file created from the DMPQ operation (the data set name is found in the UCC7QDMP DD statement of CA7ONL). This is used as input to the queue renames processor (Step 8).

4. Create a sequential backup of the CA Workload Automation SE database using IDCAMS or SASSBK00.
5. Optionally, create a sequential backup of the CA Workload Automation SE VRM file using IDCAMS.
6. Optionally, create a sequential backup of the CA Workload Automation SE ARF file using IDCAMS.
7. Tailor and run the database rename processor, CAL2JRXD, to create an altered sequential backup of the CA Workload Automation SE database.
8. Tailor and run the queue rename processor, CAL2JRXQ, to create an altered sequential backup of the CA Workload Automation SE queues. The output file from this step is used in the MOVQ operation (Step 14).
9. Optionally, tailor and run the VRM rename processor, CAL2JRXV, to create an altered sequential backup of the CA Workload Automation SE VRM database.
10. Optionally, tailor and run the ARF rename processor, CAL2JRXA, to create an altered sequential backup of the CA Workload Automation SE ARF database.
11. Reload the CA Workload Automation SE database with the altered sequential backup (created in Step 7) using IDCAMS or SASSBK00.
12. Optionally, reload the CA Workload Automation SE VRM file with the altered sequential backup (created in Step 9) using IDCAMS. To reload the VRM database, you may choose to use CA07N516 job in JCLLIB or the JCL procedure CA7VRL.
13. Optionally, reload the CA Workload Automation SE ARF file with the altered sequential backup (created in Step 10) using IDCAMS. To reload the ARF database, you may choose to use CA07N518 job in JCLLIB or the JCL procedure CA7ARL.
14. Start CA Workload Automation SE using the MOVQ option to reload the queues from the altered sequential backup (created in Step 8).

    Note: If IDCAMS is used to back up the CA Workload Automation SE database, there should be three files that are used to rename jobs on the SASDS, SASJOB, and IDS data sets. Thus, Steps 4, 7, and 11 would be executed for each of the data sets forming the database. Also, if IDCAMS is used, when reloading the database and/or VRM database, ensure that the data set has been deleted and redefined. If not, IDCAMS issues "duplicate keys" messages during the reload.
15. Review global variables, if any. Check names, values, and masks to determine whether any need to be changed. Use the /GVAR command to either modify global variables or delete and add global variables as appropriate.

Mass Job Rename Control File (RENAMES)

The RENAMES control file defines the job names to be changed with their corresponding new names. The same RENAMES control file should be used for all rename processors to ensure that the changes are consistent across all elements.

The RENAMES control file is a physical sequential file or member with 80-character, fixed length records. The block size should be a multiple of 80. It can be a sequential file, or a member in a PDS. The format of RENAMES control file records is:

Column 1 - 8 : Old job name
Column 9 : blank
Column 10 - 17 : New job name
Column 18 : blank
Column 19 - 80 : comment area

If Column 9 is not a blank (space), the statement is treated as a comment statement and is ignored.

You can have comments within the RENAMES file by using a blank (space) or an asterisk (*) in column one. Comment statements are bypassed and are not echoed in any reports.

If there are any problems with the RENAMES file, a U0001 abend is produced when executing any of the utilities.

Database Rename Process (CAL2JRXD)

The CA Workload Automation SE database rename processor, CAL2JRXD, handles elements in the CA Workload Automation SE job, dataset, and IDS files. It runs against an IDCAMS or SASSBK00 sequential backup of the CA Workload Automation SE database. The output from the process is an altered sequential backup of the CA Workload Automation SE database. If any job names are changed, the altered backup must be sorted before it can be used to reload the CA Workload Automation SE VSAM database files.

If IDCAMS is used to create the CA Workload Automation SE database backup, each data set comprising the database, SASDS, SASJOB, and IDS, should be backed up to a separate file. Then, use the CAL2JRXD utility against each of those backup files. When reloading the database, each newly created file should be reloaded to a newly defined data set.

CAL2JRXD Process JCL

The following JCL can be used to run the CAL2JRXD process. A copy of this JCL can be found in member AL2ZJRND in the CA Workload Automation SE JCL library (CAL2JCL).

```
//JOBX      JOB  .................
//*********************************************************************
//*           CA 7 MASS JOB RENAME (DATA BASE)
//*********************************************************************
//JRXD1    EXEC PGM=CAL2JRXD
//STEPLIB  DD   DISP=SHR,DSN=cai.CAL2LOAD
//SYSUDUMP DD   SYSOUT=*
//SUMMARY  DD   SYSOUT=*,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=133)
//SYSPRINT DD   SYSOUT=*,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=133)
//*--------------------------------------------------------------------
//* DBIN IS THE SEQUENTIAL DATA BASE BACKUP INPUT (IDCAMS OR SASSBK00)
//* DBOUT IS THE SEQUENTIAL DATA BASE BACKUP OUTPUT (CONVERTED)
//*          TO JUST GET THE CHANGE REPORT KEEP DBOUT DUMMIED.
//*--------------------------------------------------------------------
//DBIN     DD   DISP=SHR,DSN=...SASSBK00 OR IDCAMS BACKUP...
//DBOUT    DD   DISP=(NEW,PASS),DSN=&&CHANGED,
//           UNIT=SYSDA,SPACE=(CYL,(50,50),RLSE),
//           DCB=(RECFM=VB,LRECL=4096,BLKSIZE=23476)
//*--------------------------------------------------------------------
//* RENAMES : CONTROL CARD INPUT TO SPECIFY JOB NAMES TO BE CHANGED
//*--------------------------------------------------------------------
//RENAMES  DD   *
   THIS LINE IS A COMMENT SINCE COLUMN ONE IS BLANK
OLDNAME1 NEWNAME1
OLDNAME2 NEWNAME2
...ETC.. ..ETC...
/*
```

```
//*-------------------------------------------------------------***
//* SORT CONVERTED CA 7 DATA BASE BACKUP AFTER MASS RENAME
//*-------------------------------------------------------------***
//SORT   EXEC  PGM=SORT,PARM='CORE=MAX,BIAS=1000'
//SORTIN   DD  DISP=(OLD,PASS),DSN=&&CHANGED
//SORTOUT  DD  DSN=...CONVERTED.AND.SORTED.BACKUP...,
//           DISP=(NEW,CATLG,DELETE),
//           UNIT=SYSDA,SPACE=(CYL,(50,50),RLSE),
//           DCB=(RECFM=VB,LRECL=4096,BLKSIZE=23476)
//SYSIN    DD  *
    SORT   FIELDS=(5,60,CH,A)
    RECORD TYPE=VB,LENGTH=(4096,,,80,320)
/*
//SYSOUT   DD  SYSOUT=*
//SORTWK01 DD  UNIT=SYSDA,SPACE=(CYL,10,,CONTIG)
//SORTWK02 DD  UNIT=SYSDA,SPACE=(CYL,10,,CONTIG)
//SORTWK03 DD  UNIT=SYSDA,SPACE=(CYL,10,,CONTIG)
//SORTWK04 DD  UNIT=SYSDA,SPACE=(CYL,10,,CONTIG)
//SORTWK05 DD  UNIT=SYSDA,SPACE=(CYL,10,,CONTIG)
//SORTWK06 DD  UNIT=SYSDA,SPACE=(CYL,10,,CONTIG)
//
```

CAL2JRXD Sample Reports

The following are the CAL2JRXD sample reports:

```
CAL2JRXD-01                          C A - 7   M A S S   J O B   R E N A M E                          PAGE    1
 DATE: dd mon yy / yy.ddd                 D A T A B A S E   U T I L I T Y

                                      J O B   R E N A M E   C H A N G E   L O G

   Sequence#   Type SubType   Old-Name  New-Name   Processing Note

         150   DSM            BD44TST1  BD22TST1   Using for DS00000001
         152   DSM            BD44BD9X  BD22BD9X   Using for DS00000001
         481   I    JDEP.     BD44BD02  BD44BD01
         481   I    JDEP.     BD44BD02  BD44BD01   List for  BD44BD01
         482   I    JDEP.     BD44BD09  BD22BD09
         804   I    PP.       BD44BD01  BD22BD01   Job Prose
       1,136   I    TRGD.     BD44BD09  BD22BD09   List for  BD22BD9X
       1,137   I    TRGD.     BD44SELF  BD22SELF
       1,138   I    TRGD.     BD44XE04  BD22XE04
       6,303   JBD            BD44      BD22
       6,304   JBD            BD44AGNT  BD22AGNT
       6,305   JBD            BD44AGN2  BD22AGN2
      22,841   JBM            BD44      BD22
```

```
CAL2JRXD-02                          C A - 7   M A S S   J O B   R E N A M E                          PAGE    1
 DATE: dd mon yy / yy.ddd                 D A T A B A S E   U T I L I T Y

                                        R E C O R D   T Y P E   S U M M A R Y

 DATA                                    INPUT     NUMBER
 TYPE   DESCRIPTION-------------        RECORDS    CHANGED

 CNTL - CONTROL RECORD                        3
 DSD  - DATASET DIRECTORY                   143
 DSM  - DATASET ENTRY                       192         38
 I    - INDEX ENTRY                       5,472         10
 JBD  - JOB DIRECTORY                    11,638         61
 JBM  - JOB RECORDS:
          JOB ENTRY                      11,638         61
          STEP ENTRY                        908         60
          DD ENTRY                        2,622        333
          REQUIREMENT                    59,267        446
          NETWORK REQUIREMENT                13
          USER REQUIREMENT                   47
          REPORT ENTRY                        0
          XPJOB ENTRY                       793
          AGENT ENTRY                        40
```

CAL2JRXD Specific Element Changes

The CA Workload Automation SE database contains a variety of records. Many of these records reference job names as part of the key or in the record because the job is a trigger or predecessor/requirement. Those fields that are defined as containing job names are changed, if necessary. Other data areas, such as text within Prose members, may happen to contain job names. These areas are not changed with this utility. If you have specific job names mentioned in the prose, these job names are *not* changed to new names.

Note: The one exception to this rule is the JCL Member field of the job definition panel. If the JCL Member name exactly matches the job name, it is changed if the job name is being changed. If the JCL member name does not exactly match the job name, it is not altered, even if the job name is changed.

Error Conditions

Any warning has a WTO written to the job's JES log and return code four (RC4) issued when the program terminates. Any error encountered also has a WTO written to the job's JES log just before issuing a user abend. If the program does abend or terminates with RC4, check the message and take corrective action.

U0001

Indicates a problem occurred with the RENAMES file.

U0002

Indicates a problem occurred in the Database Job Rename utility, such as a missing DD statement.

Queue Rename Process (CAL2JRXQ)

The CA Workload Automation SE queue rename processor, CAL2JRXQ, handles elements in the CA Workload Automation SE queue files, including the request, ready, active and prior run queues. It runs against a sequential backup of the queues generated by a DMPQ shutdown of CA Workload Automation SE. The output from the process is an altered sequential backup of the CA Workload Automation SE queue records. The output from CAL2JRXQ does not need to be sorted before it is used as input to a CA Workload Automation SE MOVQ startup.

Because the CA Workload Automation SE online system uses the file, you should make a backup copy of the DMPQ file for input to the rename process. Output from the rename process is the file name as stated in the UCC7QDMP DD statement in CA7ONL that the MOVQ process uses during CA Workload Automation SE startup. The backup copy of the DMPQ file can be used in the unlikely event that the database and queues need to be restored to their original state.

CAL2JRXQ Process JCL

The following sample JCL can be used to execute the CAL2JRXQ utility. A copy of this JCL can be found in member AL2ZJRNQ in the CA Workload Automation SE JCL library (CAL2JCL).

```
//JOBQ      JOB  .................
//*********************************************************************
//*           CA 7 MASS JOB RENAME (QUEUES)
//*********************************************************************
//JRXQ1    EXEC PGM=CAL2JRXQ
//STEPLIB  DD   DISP=SHR,DSN=cai.CAL2LOAD
//SYSUDUMP DD   SYSOUT=*
//SUMMARY  DD   SYSOUT=*,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=133)
//SYSPRINT DD   SYSOUT=*,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=133)
//*--------------------------------------------------------------------
//* QDUMPIN IS THE SEQUENTIAL BACKUP INPUT (CA 7 SHUTDOWN WITH DMPQ)
//* QDUMPOUT IS THE SEQUENTIAL QUEUE BACKUP OUTPUT (CONVERTED)
//*      TO JUST GET THE CHANGE REPORT KEEP QDUMPOUT DUMMIED.
//* RENAMES : CONTROL CARD INPUT TO IDENTIFY JOB NAMES TO BE CHANGED
//*--------------------------------------------------------------------
//QDUMPIN  DD   DISP=SHR,DSN=...QUEDMP FROM CA 7 SHUTDOWN...
//QDUMPOUT DD   DUMMY,DCB=(RECFM=VB,LRECL=256,BLKSIZE=6148)
//*QDUMPOUT DD  DSN=...QDMP FILE NAME OF CA 7...,
//*            DISP=(SHR,KEEP,KEEP)              <--- EXISTING FILE
//RENAMES  DD   *
   THIS LINE IS A COMMENT SINCE COLUMN ONE IS BLANK
OLDNAME1 NEWNAME1
OLDNAME2 NEWNAME2
...ETC.. ..ETC...
/*
```

CAL2JRXQ Sample Reports

The following are the CAL2JRXQ sample reports:

```
CAL2JRXQ-01                          C A - 7   M A S S   J O B   R E N A M E                          PAGE   1
DATE: dd mon yy / yy.ddd                        Q U E U E   U T I L I T Y

                                    J O B   R E N A M E   C H A N G E   L O G

  Sequence#   Type SubType   Old-Name  New-Name   Processing Note

        488   PRNQ HEADER    BD44BD01  BD22BD01
        504   PRNQ HEADER    BD44AGN2  BD22AGN2
        505   PRNQ HEADER    BD44SELF  BD22SELF
        510   PRNQ HEADER    BD44TAN1  BD22TAN1
        511   PRNQ HEADER    BD44BD02  BD22BD02
        512   PRNQ HEADER    BD44BD03  BD22BD03
        554   PRNQ HEADER    BD44ARF1  BD22ARF1
        559   PRNQ HEADER    BD44BD1X  BD22BD1X
        573   PRNQ HEADER    BD44BD9X  BD22BD9X
        574   PRNQ HEADER    BD44PH01  BD22PH01
        576   PRNQ HEADER    BD44XE04  BD22XE04
        582   PRNQ HEADER    BD44BD04  BD22BD04
        584   PRNQ HEADER    BD44      BD22
```

```
CAL2JRXQ-02                          C A - 7   M A S S   J O B   R E N A M E                          PAGE     1
DATE: dd mon yy / yy.ddd                        Q U E U E   U T I L I T Y

                                      R E C O R D   T Y P E   S U M M A R Y

DATE/TIME FROM CONTROL RECORD ON QDUMPIN: yyyy.ddd  hh:mm:ss.th

Data                                  Input     Number
Type   Description--------------     Records    Changed
CNTL - CONTROL RECORD (START)              1
CNTL - CONTROL RECORD (WINDUP)             0
PREQ - HEADER                              0
PREQ - JCL                                 0
REQQ - HEADER                            447
REQQ - LOAD                                0
REQQ - XPJOB                              14
REQQ - AGENT                               1
REQQ - JCL                                35
REQQ - INPUT                               1
REQQ - OUTPUT                              0
REQQ - SECURITY                            0
REQQ - #SCC INPUT                          0
.....
ACTQ - #SCC INPUT                          0
ACTQ - #SCC OUTPUT                         0
ACTQ - ARF                                 0
ACTQ - XPS                                 0
PRNQ - HEADER                            393         33
PRNQ - LOAD                                0
REQQ - XPJOB                              20
REQQ - AGENT                               0
PRNQ - JCL                               390
PRNQ - INPUT                               0
PRNQ - OUTPUT                              0
.....
UNIDENTIFIED RECORD TYPES                  0
       TOTALS                          1,305         33
```

Specific Element Changes

When a DMPQ operation has been executed, the pre- and post-processing, request, ready, active, and prior-run queues are written to a sequential data set. *Only job names in the prior-run queue records are changed to a new name*. The request, ready, and active queue records reflect jobs that are already active in the system and will not be renamed. However the next time the newly named job is scheduled, it is submitted as the new name. It is important that the prior-run records are renamed, since those records are used for job requirement satisfaction.

Error Conditions

Any warning has a WTO written to the job's JES log and return code four (RC4) issued when the program terminates. Any error encountered also has a WTO written to the job's JES log just before issuing a user abend. If the program does abend or terminates with RC4, check the message and take corrective action.

U0001

Indicates a problem occurred with the RENAMES file.

U0003

Indicates a problem occurred in the Queue Job Rename utility, such as a missing DD statement.

VRM Process (CAL2JRXV)

The CA Workload Automation SE VRM rename processor, CAL2JRXV, handles elements in the CA Workload Automation SE VRM file. It should be used if you use the CA Workload Automation SE Virtual Resource Management (VRM) feature of CA Workload Automation SE. It runs against an IDCAMS sequential backup of the CA Workload Automation SE VRM file. The output from the process is an altered sequential backup of the CA Workload Automation SE VRM file. If any job names are changed the altered backup must be sorted before it can be used to reload the CA Workload Automation SE VRM VSAM file.

When reloading the VRM database, the JCLLIB member CA07N516, which invokes JCL procedure CA7VRL, can be used to delete, define, and load the output from the CAL2JRXV utility.

Global variables are also changed with the execution of CAL2JRXV. The CAL2JRXV program looks for a specific job name within GVAR definition, and if a specific job name matches a job name being renamed, job name renaming occurs.

CAL2JRXV Process JCL

The following JCL can be used to run the CAL2JRXV process. A copy of this JCL can be found in member AL2ZJRNV in the CA Workload Automation SE JCL library (CAL2JCL).

```
//JOBX      JOB  .................
//*********************************************************************
//*           CA 7 MASS JOB RENAME (VRM)
//*********************************************************************
//JRXD1    EXEC PGM=CAL2JRXV
//STEPLIB  DD   DISP=SHR,DSN=cai.CAL2LOAD
//SYSUDUMP DD   SYSOUT=*
//SUMMARY  DD   SYSOUT=*,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=133)
//SYSPRINT DD   SYSOUT=*,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=133)
//*--------------------------------------------------------------------
//* VRMIN IS THE SEQUENTIAL IDCAMS VRM BACKUP
//* VRMOUT IS THE SEQUENTIAL VRM BACKUP OUTPUT (CONVERTED)
//*          TO JUST GET THE CHANGE REPORT KEEP VRMOUT DUMMIED.
//*--------------------------------------------------------------------
//VRMIN     DD   DISP=SHR,DSN=...IDCAMS BACKUP...
//VRMOUT    DD   DISP=(NEW,PASS),DSN=&&CHANGED,
//           UNIT=SYSDA,SPACE=(CYL,(50,50),RLSE),
//           DCB=(RECFM=VB,LRECL=32700,BLKSIZE=32704)
//*--------------------------------------------------------------------
//* RENAMES : CONTROL CARD INPUT TO SPECIFY JOB NAMES TO BE CHANGED
//*--------------------------------------------------------------------
//RENAMES  DD   *
   THIS LINE IS A COMMENT SINCE COLUMN ONE IS BLANK
OLDNAME1 NEWNAME1
OLDNAME2 NEWNAME2
...ETC.. ..ETC...
/*
```

```
//*-------------------------------------------------------------***
//* SORT CONVERTED CA 7 VRM BACKUP AFTER MASS RENAME
//*-------------------------------------------------------------***
//SORT   EXEC  PGM=SORT,PARM='CORE=MAX,BIAS=1000'
//SORTIN   DD  DISP=(OLD,PASS),DSN=&&CHANGED
//SORTOUT  DD  DSN=...CONVERTED.AND.SORTED.BACKUP...,
//           DISP=(NEW,CATLG,DELETE),
//           UNIT=SYSDA,SPACE=(CYL,(50,50),RLSE),
//           DCB=(RECFM=VB,LRECL=4096,BLKSIZE=23476)
//SYSIN    DD  *
    SORT   FIELDS=(5,53,CH,A)
    RECORD TYPE=VB,LENGTH=(32704)
/*
//SYSOUT   DD  SYSOUT=*
//SORTWK01 DD  UNIT=SYSDA,SPACE=(CYL,10,,CONTIG)
//SORTWK02 DD  UNIT=SYSDA,SPACE=(CYL,10,,CONTIG)
//SORTWK03 DD  UNIT=SYSDA,SPACE=(CYL,10,,CONTIG)
//SORTWK04 DD  UNIT=SYSDA,SPACE=(CYL,10,,CONTIG)
//SORTWK05 DD  UNIT=SYSDA,SPACE=(CYL,10,,CONTIG)
//SORTWK06 DD  UNIT=SYSDA,SPACE=(CYL,10,,CONTIG)
//
```

CAL2JRXV Sample Reports

The following are the CAL2JRXV sample reports:

```
CAL2JRXV-01                          C A - 7   M A S S   J O B   R E N A M E                          PAGE   1
DATE: dd mmm yy / yy.ddd                        V R M   U T I L I T Y

                                   J O B   R E N A M E   C H A N G E   L O G

  Sequence#   Type   Old-Name  New-Name   Processing Note
         20     J     A00001    Z00001
         36     J     BDFLOW1   FLOWERO1
         36     J     BDFLOW3   FLOWERO3   Job FLOWERO1 Flow BDFLOW1B end job changed
         36     J     BDFLOW3   FLOWERO3   Job FLOWERO1 Flow BDFLOW1A end job changed
         46     J     BD44      BD22
         55     J     FLOW1A    FLOWER1A
         56     J     FLOW2D1   FLOWERD1   Job FLOW2A1  Flow TFLOW2   end job changed
         56     J     FLOW2B2   FLOWER2B   Job FLOW2A1  Flow XFLOW1   end job changed
         56     J     FLOW2D1   FLOWERD1   Job FLOW2A1  Flow TFLOW11  end job changed
         56     J     FLOW2D1   FLOWERD1   Job FLOW2A1  Flow TFLOW12  end job changed
        155     R     A00001    Z00001     Resource A00001.CRQ
        178     R     BDFLOW1   FLOWERO1   Resource FLOW@BDFLOW1A.BDFLOW3.001.1230
        178     R     BDFLOW3   FLOWERO3   Job FLOWERO1 Flow BDFLOW1A end job changed
        179     R     BDFLOW1   FLOWERO1   Resource FLOW@BDFLOW1B.BDFLOW3.002.1600
        179     R     BDFLOW3   FLOWERO3   Job FLOWERO1 Flow BDFLOW1B end job changed
        187     R     FLOW1A    FLOWER1A   Resource FLOW@TFLOW1.FLOW3C.1.2330.1
        190     R     FLOW2D1   FLOWERD1   Job FLOW2A1  Flow TFLOW11  end job changed
        195     Z     BDFLOW1   FLOWERO1   GVAR VARIABLE                      System TESTSYS  SchID 001
```

```
CAL2JRXV-02                          C A - 7   M A S S   J O B   R E N A M E                          PAGE   1
 DATE: dd mon yy / yy.ddd                          V R M   U T I L I T Y

                                        R E C O R D   T Y P E   S U M M A R Y

DATA                                                    INPUT     NUMBER
TYPE   DESCRIPTION-----------------------              RECORDS   CHANGED
 F    CA-7 Active CPM Records                              10
 J    CA-7 VRM Job                                         72         5
 N    CA-7 XPJOB Node                                      17
 O    CA-7 XPJOB Owner                                      5
 P    CA-7 User Profile                                    49
 R    CA-7 VRM Resource XREF                              128         9
 U    CA-7 VRM Manually Freed RCT                           0
 V    CA-7 VRM Manually Freed Resource                      0
 W    CA-7 VRM Corequisite Resource                         4
 X    CA-7 VRM Active Job                                   1
 Y    CA-7 VRM Active RCT Resource                         40
 Z    CA-7 Variable Record                                  5         1
 9    CA-7 VRM Database Init Record                         1

      TOTALS                                              338        15
```

Specific Element Changes

The VRM database contains many different record types. Only the Job, Resource, and Variable (J, R, and Z) record types change with the execution of CAL2JRXV. Since the following record types do not change, you will not see any change counts associated with these record types:

- Active CPM flow (F)
- Profile (P)
- XPJOB node (N)
- XPJOB owner (O)
- Manual free resource count (U)
- Manual free resource (V)
- Co-requisite resource (W)
- Active job (X)
- Active resource count (Y)
- Database initialization record (9)

For the Job, Resource, and Variable (global variable) record types, the job name can change if the job is coded in the RENAMES files. For resource records, the ending job name found in a CPM FLOW resource name is also eligible to be changed. However, if you use job names as resource names for other types of resources, those job names are not changed.

Error Conditions

Any warning has a WTO written to the job's JES log and return code four (RC4) issued when the program terminates. Any error encountered also has a WTO written to the job's JES log just before issuing a user abend. If the program does abend or terminates with RC4, check the message and take corrective action.

U0001

Indicates a problem occurred with the RENAMES file.

U0004

Indicates a problem occurred in the VRM Job Rename utility, such as a missing DD statement.

ARF Process (CAL2JRXA)

The ARF rename processor, CAL2JRXA, handles elements in the ARF file. It should be used if you use the CA Workload Automation SE Automatic Recovery Facility (ARF) feature. It runs against an IDCAMS sequential backup of the CA Workload Automation SE ARF file. The output from the process is an altered sequential backup of the ARF file. The output file is ready for reload since no VSAM keys are changed in the ARF file.

When reloading the ARF database, the JCLLIB member CA07N518, which invokes JCL procedure CA7ARL, can be used to delete, define, and load the output from the CAL2JRXA utility.

CAL2JRXA Process JCL

The following JCL can be used to run the CAL2JRXA process. A copy of this JCL can be found in member AL2ZJRNA in the CA Workload Automation SE JCL library (CAL2JCL).

```
//JOBX      JOB  .................
//*********************************************************************
//*           CA 7 MASS JOB RENAME (ARF)
//* USE THIS MODEL JCL TO CHANGE LARGE NUMBERS OF JOBNAMES.  JOBNAMES
//* WILL BE CHANGED IN EVERY ARF ELEMENT WHERE THEY APPEAR.
//* INPUT TO THE PROCESS IS A SEQUENTIAL ARF FILE BACKUP.
//* OUTPUT FROM THE PROCESS IS AN UPDATED ARF BACKUP SUITABLE FOR
//* RELOADING.
//*********************************************************************
//JRXV1    EXEC PGM=CAL2JRXA
//STEPLIB  DD   DISP=SHR,DSN=cai.CAL2LOAD
//SYSUDUMP DD   SYSOUT=*
//*--------------------------------------------------------------------
//* SUMMARY WILL HAVE THE PROCESS CONTROL REPORT WITH TOTALS.
//* SYSPRINT WILL HAVE THE PROCESSING REPORT DETAILING
//* INDIVIDUAL RECORD CHANGES AND A RECORD CHANGE SUMMARY.
//*--------------------------------------------------------------------
//SUMMARY  DD   SYSOUT=*,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=133)
//SYSPRINT DD   SYSOUT=*,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=133)
//*--------------------------------------------------------------------
//* ARFIN IS THE SEQUENTIAL IDCAMS ARF BACKUP INPUT
//*--------------------------------------------------------------------
//ARFIN    DD   DISP=SHR,DSN=...IDCAMS BACKUP...
```

```
//*-------------------------------------------------------------------
//* ARFOUT IS THE SEQUENTIAL ARF BACKUP OUTPUT (CONVERTED);
//*      TO JUST GET THE CHANGE REPORT KEEP ARFOUT DUMMIED.
//* TO CREATE THE OUTPUT FILE, USE THE SAME SIZE AND
//*      DCB CHARACTERISTICS AS ARFIN ABOVE; CHECK THE SPACE PARM.

//*-------------------------------------------------------------------
//*RFOUT    DD   DUMMY,DCB=(RECFM=VB,LRECL=32700,BLKSIZE=32704)
//ARFOUT   DD  DSN=...CONVERTED.ARF.OUPUT...,
//           DISP=(NEW,CATLG,DELETE),
//           UNIT=SYSDA,SPACE=(CYL,(50,50),RLSE),
//           DCB=(RECFM=VB,LRECL=32700,BLKSIZE=32704)
//*-------------------------------------------------------------------
//* RENAMES IS THE CONTROL CARD INPUT TO SPECIFY JOB NAMES TO BE
//* CHANGED.  INPUT CAN BE INSTREAM (LIKE SYSIN), OR THE DD CAN
//* POINT TO A DATA SET OR PDS MEMBER (AS LONG AS LRECL=80).
//* ANY LINES THAT START WITH A BLANK OR ASTERISK ARE COMMENTS.
//*
//* COLUMNS  1 -  8      OLD JOB NAME
//* COLUMN     9           BLANK
//* COLUMNS 10 - 17      NEW JOB NAME
//* COLUMN    18           BLANK
//* COLUMNS 19 - 80      COMMENT AREA
//*
//* RIGHT NOW YOU CANNOT USE ANY GENERICS OR WILDCARDS.
//*-------------------------------------------------------------------
//RENAMES  DD   *
   THIS LINE IS A COMMENT SINCE COLUMN ONE IS BLANK
OLDNAME1 NEWNAME1
OLDNAME2 NEWNAME2
...ETC.. ..ETC...
/*
//
```

CAL2JRXA Sample Reports

The following are the CAL2JRXA sample reports:

```
CAL2JRXA-01                          C A - 7   M A S S   J O B   R E N A M E                          PAGE   1
DATE: DD MMM YY / YY.JJJ                        A R F   U T I L I T Y
                                              J O B   R E N A M E   C H A N G E   L O G
  Sequence#    ARFSET     Def ID  Resp#   Type    Old name   New Name
          1    EROCARFZ       2    1       AJ     CLEN       CA07CLEN
          1    EROCARFZ       3    1       AC     CLEN       CA07CLEN
          2    EROCARF1      20    2       AJ     CLEN       CA07CLEN
          3    GUESTER1      10    1       AJ     CLEN       CA07CLEN
          3    GUESTER1      10    2       AC     EOCX01     PREFXX01
          3    GUESTER1      10    2       AC     CLEN       CA07CLEN
          3    GUESTER1      20    2       AC     EOCX01     PREFXX01
          3    GUESTER1      20    2       AC     CLEN       CA07CLEN
          4    TESTEC1      100    2       AC     CLEN       CA07CLEN
          4    TESTEC1      100    4       AC     EOCX01     PREFXX01
          5    TESTEE1      100    2       AC     CLEN       CA07CLEN
          5    TESTEE1      100    4       AC     EOCX01     PREFXX01
          5    TESTEE1      101    4       AC     EOCX01     PREFXX01
          5    TESTEE1      101    5       AC     CLEN       CA07CLEN
```

```
CAL2JRXA-02                          C A - 7   M A S S   J O B   R E N A M E            PAGE   1
DATE: DD MMM YY / YY.JJJ                        A R F   U T I L I T Y
                                              R E C O R D   T Y P E   S U M M A R Y
DATA                                             INPUT     NUMBER
TYPE   DESCRIPTION-----------------------      RECORDS    CHANGED
 AA    CA-7 Data Base ARF Records                    6          6
 AB    CA-7 Active ARF Records                       0
 AQ    CA-7 Active Queue ARF Records                 1
 99    CA-7 ARF Initialization Records               1

       TOTALS                                        8          6
```

Specific Element Changes

In the ARF database, ARFSETs contain various definition IDs (DEFID) and the responses for that DEFID. Each DEFID within an ARFSET is examined, and for any ARF JOB (AJ) or ARF Command (AC) response, the job names are examined to see whether the name matches one in the RENAMES DD list. For the AJ replacement, it is a straight job name replacement. For the AC replacement, the command is scanned for JOB=*jobname*. This can appear as keyword JOB=, DEPJOB=, and other variations. If a match is found, the program verifies that larger names can fit in the allotted space (for the command). If the new name fits, the replacement is performed. If the new name expands the command so that it exceeds the command area, an error message is written in the report, and a return code four (RC4) is issued when the program terminates.

Error Conditions

If a command within an ARFSET is trying to be expanded beyond the 69-byte limit, the CAL2JRXA program issues a message. The message is displayed in the CAL2JRXA-01 report and shows the command that could not be changed. This situation usually occurs when trying to rename a smaller name to a larger name. The message appears below the line that states the attempted rename and is as follows:

*WARNING: Command *NOT* changed; new cmd too long. *Command echoed here*

Any warning has a WTO written to the job's JES log and return code four (RC4) issued when the program terminates. Any error encountered also has a WTO written to the job's JES log just before issuing a user abend. If the program does abend or terminates with RC4, check the message and take corrective action.

U0001

Indicates a problem occurred with the RENAMES file.

U0002

Indicates a problem occurred in the ARF Job Rename utility, such as a missing DD statement.

Appendix A: Using CA 7/PERSONAL SCHEDULING

This section contains the following topics:

Overview (see page 473)
Panel Descriptions (see page 474)
Access CA-7/PERSONAL SCHEDULING (see page 477)
Add a Job (see page 479)
Submit a Job (see page 480)
Display a Job's Status (see page 481)
Implement CA 7/PS (see page 482)
CA 7/PS Messages (see page 484)

Overview

CA 7/PERSONAL SCHEDULING (CA 7/PS) is designed to extend the functionality of scheduling jobs to virtually anyone within a company who has access to a terminal connected to their mainframe. It is an easy way to set up and monitor jobs without having to spend much time learning to use the various extended functions that are available in CA Workload Automation SE.

Important! CA 7/PS was designed only for CPU job types; however all options except ADD (define) can be used for any job type (for example, CPU or cross-platform). If you want to ADD a job, it is automatically added as a CPU job. You cannot ADD an XPJOB or an agent job using the PS panel.

CA 7/PS provides the following functions:

- Add a job to CA Workload Automation SE
- Schedule the job to run on a date/time basis
- Run a one-time job without having to add it to CA Workload Automation SE
- Run a job in sequence with another job
- Display the last time that a job ran
- Display the next time that the job is scheduled
- Display information about a job that is running (or about to run)
- Delete previously defined jobs

Panel Descriptions

The following is a CA-7/PERSONAL SCHEDULING panel.

```
------------------------- CA-7 / PERSONAL SCHEDULING ------ yy.ddd / hh:mm:ss

  FUNCTION===>                (ADD,LIST,UPD,DEL,SUBMIT,STATUS,QUIT)

                     JOBNAME =>
                      JCL ID =>       JCL LIB =>

               WHEN TO SCHEDULE:
                  PATTERN     =>
                  TIME TO RUN =>           (0001-2400)

               JOB RUNS AFTER =>           (JOBNAME - SUBMIT ONLY)


         - - - - - - -  STATUS INFORMATION (DISPLAY ONLY)  - - - - - - -




 MESSAGE:  PS00-00 ENTER THE DESIRED FUNCTION AND JOB INFORMATION.
```

This panel contains the following fields:

FUNCTION

Determines what action is taken when Enter is pressed. Available FUNCTIONS are:

ADD

Adds the job to CA Workload Automation SE. Note the following defaults (using the CA-7 CPU Job Definition panel for field names): the JOBNET field defaults to the PATTERN name if one is entered, the SYSTEM field defaults to CA7PS, the OWNER field is the USERID of the person adding the job and the UID is the UID of the person adding the job. All other defaults are taken from the Personal Scheduling or regular DEFAULTS job as defined at your installation.

LIST

Lists the information that has previously been defined to CA Workload Automation SE using the ADD and/or UPD functions.

UPD

Updates the CA Workload Automation SE information about the job with the information provided in the other fields on this panel. If external security is used, the person updating must have the same USERID as the OWNER field on the JOB panel and the same UID (coids are not honored).

DEL

Deletes all of the job's information from CA Workload Automation SE.

SUBMIT

Submits the job listed in the JOBNAME field to the CPU. It runs without waiting for the scheduled date/time. The PATTERN field is not honored with this function. This function can be used to run a job without having to set up a PATTERN and TIME for the job to be scheduled. The TIME field may be used to establish a submission time of day within the next 24 hours. The JOB RUNS AFTER field can be used to cause the job to wait on the execution of another job that runs under CA Workload Automation SE control. To add the job to the database, the ADD function must be used.

STATUS

Displays the current information about the job. This may include information about when the job last ran, when it is scheduled to run next, and current status if the job is executing or waiting to be submitted. The information is displayed on the lower half of the panel, below the STATUS INFORMATION line. These are the possible messages that can be displayed:

- JOB IS NEXT SCHEDULED AT *hhmm* ON *yy.ddd*
- JOB LAST RAN AT *hhmm* ON *yy.ddd*
- JOB IS WAITING FOR JOB *xxxxxxxx*
- JOB IS AWAITING EXECUTION
- JOB IS ACTIVE ON CPU
- JOB DOES NOT CURRENTLY HAVE A SCHEDULE
- JOB IS NOT DEFINED IN THE CA-7 DATABASE

QUIT

Terminates the current session with CA Workload Automation SE.

JOBNAME

Indicates the job on which the indicated function will be performed. The field may be 1-8 alphanumeric characters. Required.

JCL ID

Identifies the JCL data set which contains the execution JCL to be submitted. If used, the value must be a numeric INDEX associated with the desired JCL data set (on the JCL statement in the initialization file). The numeric INDEX specified must be a number between 0 and 253. Refer to the *Systems Programming Guide* for further information on the initialization file. This field or the JCL LIB field is required if the job is not defined in the database. JCL ID and JCL LIB are mutually exclusive.

JCL LIB

Identifies the JCL data set which contains the execution JCL to be submitted. If used, the value must be a symbolic INDEX associated with the desired JCL data set (on the JCL statement in the initialization file). Refer to the *Systems Programming Guide* for further information on the initialization file. This field or the JCL ID field is required if the job is not defined in the database. JCL ID and JCL LIB are mutually exclusive.

PATTERN

Indicates the days that this job is going to run. Each site has a unique set of PATTERN jobs defined that give you flexibility in scheduling your jobs. Optional.

If PATTERN is input and it is the same as the one already defined, then no changes are made to the job's schedule, even if the schedule of the PATTERN job has changed. To delete a PATTERN, enter =D in the PATTERN field and use the UPD function.

TIME TO RUN

Indicates the time of the day(s) that this job is submitted. This value may be specified from 0001 (one minute after midnight) through 2400 (midnight). Make sure that you specify the correct hour (3:00 p.m. is actually specified as 1500). Required if PATTERN is used.

TIME may also be used with the SUBMIT function to demand a job to be submitted in the next 24 hours. If the time of day specified is less than the current time of day, the job is submitted at that time on the following day.

JOB RUNS AFTER

Used to indicate that the job you are submitting must wait for another job to complete before it can run. This allows you to establish sequencing for the job. This field is only valid with the SUBMIT function. Optional.

Access CA-7/PERSONAL SCHEDULING

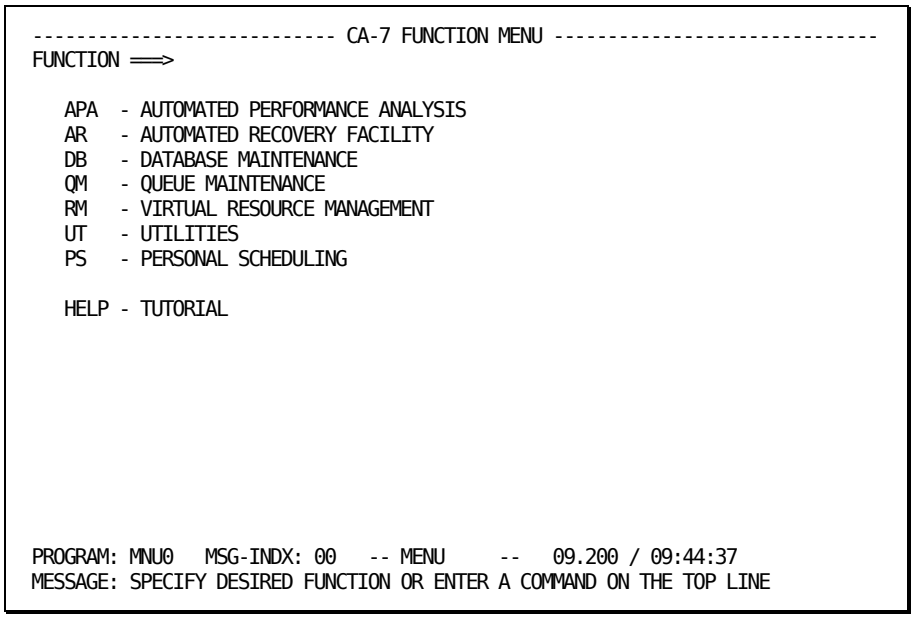
To access CA-7/PS:

1. Log on to CA Workload Automation SE. This can be accomplished several ways. CA Workload Automation SE may be accessed directly from your site's VTAM panel, through a session manager, or through TSO/ISPF.

   When you have logged on to CA Workload Automation SE successfully, a panel similar to the one below is displayed.

```
-----------------------*** CA71 PRODUCTION INSTANCE ***---------------------

PLEASE ENTER LOGON DATA OR PRESS PF3 TO DISCONNECT



USERID       :              TERMINAL NAME : TRM001        DATE  : yy.131
PASSWORD     :              VTAM APPLID   : CAW           TIME  : 09:01:24
NEW PASSWORD :              LUNAME        : A99L100       LEVEL : r11.3(nnnn)
UID RESOURCE :
PARMS        :



               C A   W O R K L O A D   A U T O M A T I O N




                          Copyright (c) 2009 CA.
                           All rights reserved.
```

2. Press Enter to display the following CA-7 Function Menu.



3. Type PS from the CA-7 Function Menu. Press Enter.

   The CA-7/PERSONAL SCHEDULING panel is now displayed, and you are ready to perform any of the CA-7/PS functions.

Note: The PS command can also be used at the top line of any panel to display the CA-7/PS panel.

Add a Job

```
------------------------  CA-7 / PERSONAL SCHEDULING ------ yy.ddd / hh:mm:ss

FUNCTION===>  ADD            (ADD,LIST,UPD,DEL,SUBMIT,STATUS,QUIT)

                     JOBNAME => MYJOB
                       JCL ID => 1     JCL LIB =>

               WHEN TO SCHEDULE:
                  PATTERN     => DAILY
                  TIME TO RUN => 0800      (0001-2400)

               JOB RUNS AFTER =>           (JOBNAME - SUBMIT ONLY)


      - - - - - - - STATUS INFORMATION (DISPLAY ONLY) - - - - - - -









  MESSAGE: PS00-00  ENTER THE DESIRED FUNCTION AND JOB INFORMATION.
```

This panel shows the information that is used to schedule a job. The fields that are filled in are:

- FUNCTION
- JOBNAME
- JCL ID or JCL LIB
- PATTERN
- TIME TO RUN

The information supplied will cause the job to be submitted on a daily basis at 8:00 a.m.. The PATTERN name of DAILY may not be defined at your site, but there is probably an equivalent one set up. When the job is submitted, the JCL will be copied from the JCL library defined at your site with an index of 1.

Submit a Job

```
------------------------  CA-7 / PERSONAL SCHEDULING ------ yy.ddd / hh:mm:ss

FUNCTION===>  SUBMIT         (ADD,LIST,UPD,DEL,SUBMIT,STATUS,QUIT)

                     JOBNAME => MYJOB2
                       JCL ID => 1     JCL LIB =>

               WHEN TO SCHEDULE:
                  PATTERN     =>
                  TIME TO RUN =>           (0001-2400)

               JOB RUNS AFTER => MYJOB     (JOBNAME - SUBMIT ONLY)


      - - - - - - - STATUS INFORMATION (DISPLAY ONLY) - - - - - - -









    MESSAGE: PS00-00  THE SUBMIT WAS SUCCESSFUL.
```

This panel shows the information that is used to run a job. The fields that are filled in are:

- FUNCTION
- JOBNAME
- JCL ID or JCL LIB
- JOB RUNS AFTER

The information supplied will cause the job to be submitted by CA Workload Automation SE without having to wait for scheduling by date/time criteria. The JOB RUNS AFTER field causes CA Workload Automation SE to verify that MYJOB runs before MYJOB2 will be submitted to the CPU.

Display a Job's Status

```
------------------------  CA-7 / PERSONAL SCHEDULING ------ yy.ddd / hh:mm:ss

FUNCTION===>  STATUS         (ADD,LIST,UPD,DEL,SUBMIT,STATUS,QUIT)

                     JOBNAME => MYJOB
                       JCL ID =>       JCL LIB =>

               WHEN TO SCHEDULE:
                  PATTERN     =>
                  TIME TO RUN =>           (0001-2400)

               JOB RUNS AFTER =>           (JOBNAME - SUBMIT ONLY)


      - - - - - - - STATUS INFORMATION (DISPLAY ONLY) - - - - - - -

                    - JOB IS NEXT SCHEDULED AT hhmm ON yy.ddd
                    - JOB LAST RAN AT hhmm ON yy.ddd
                    - JOB IS ACTIVE ON CPU



   MESSAGE: PS00-00  STATUS SUCCESSFUL.
```

This panel is an example of the STATUS function. The fields that are entered are:

- FUNCTION
- JOBNAME

The status indicates the next date/time that the job is going to be scheduled to run, the last date/time that it ran, and any current information if the job is in the process of being run.

Note: If the job was not added through the Personal Scheduling panel, the time on the **JOB IS NEXT SCHEDULED AT** will be all zeros.

Implement CA 7/PS

To implement CA 7/PERSONAL SCHEDULING:

1. Define the USERIDs to CA Workload Automation SE that will be using CA 7/PS.

   If CA Workload Automation SE internal security is being used, the current Security module must be modified to allow access to the PS00 application. The access levels required are shown below. For more information about the internal security module, see the *Security Reference Guide*

   If command and panel access is being controlled using the CA Workload Automation SE External Security Interface, the resource for the new panel may need to be defined to the security package. The USERIDs that will be accessing CA 7/PS must be given access to the CA Workload Automation SE application and the resource listed below, with the appropriate access level.

Panel ID	Resource Name	Access Level	Functions
PS	L2PS	READ(5)	STATUS,LIST
		ADD (10)	ADD
	UPD (10)	UPD	
	DEL (10)	DEL	
		SUBM (10)	SUBMIT
		None	QUIT

2. Set up the PATTERN jobs.

   The AL2PSBTI member of CAL2OPTN contains sample Batch Terminal commands to define several PATTERN jobs to CA Workload Automation SE. The sample commands add them to the CA Workload Automation SE database, define their schedules, and RESOLV the schedules. The following is a list of the PATTERN jobs that are defined in the CAL2OPTN member AL2PSBTI:

   DAILY

   Every available processing day of the calendar used.

   MON

   Mondays.

   TUES

   Tuesdays.

   WED

   Wednesdays.

THU

Thursdays.

FRI

Fridays.

FIRST

First day of the month (actual or relative based on the calendar used).

LAST

Last day of the month (actual or relative based on the calendar used).

Before using the AL2PSBTI member as input to the CA Workload Automation SE batch terminal, do the following:

- Verify that no jobs are already defined to the CA Workload Automation SE database with the preceding names. If there are, simply change the job names in the sample AL2PSBTI member. Note that the jobs are not going to be scheduled by CA Workload Automation SE because their schedules are turned off with the DONT SCHEDULE BEFORE field set to 99999.
- Supply valid SCAL values where SCAL is specified. The calendar (SCAL) specified will determine whether the FIRST and LAST will be actual or relative days (based on the specification of the OPTIONS=SCHDYONLY value) and whether DAILY will include Saturdays and Sundays or only weekdays.

After running a batch terminal with the sample member, verify that the commands were processed correctly without errors by examining the job's output. After this verification is complete, the installation is complete and the implementation is well under way.

3. Define Personal Scheduling Defaults Job (optional).

   You may establish a DEFAULTS job specifically for personal scheduling. If you choose not to, defaults are taken from the regular CA Workload Automation SE defaults job. For example, you may want all Personal Scheduling jobs to be MAINT=Y while your production work is normally MAINT=N. To establish a Personal Scheduling defaults job:

   - Define the job to the CA Workload Automation SE database using the job definition panel. Remember that some information (such as SYSTEM, OWNER, and UID) are always overridden when a Personal Scheduling job is ADDed.
   - In your CA Workload Automation SE initialization file, add the following keyword to the DBASE statement:

     DEFAULTPS=jobname

4. Code Personal Scheduling Verification Exit (optional).

   Refer to the *Systems Programming Guide* for information on adding and implementing an exit to allow you to enforce local standards and conventions.

Defining Additional PATTERN Jobs

If any other PATTERN streams are desired, simply define them using the batch terminal or using the CA Workload Automation SE online system. When defining any additional jobs, the following items should be considered:

- Add the jobs with EXEC=N and DONT SCHEDULE BEFORE=99999.
- Schedule the PATTERN job as desired using the CA-7 CPU JOB SCHEDULING panel. A PATTERN job may have up to 2 SCHIDs defined.
- Ensure that the schedules are RESOLVed.

Note: The PATTERN jobs should have no more than two SCHIDs defined.

CA 7/PS Messages

The following are the messages.

PS00-00

ENTER THE DESIRED FUNCTION AND JOB INFORMATION.

Reason:

This prompt requests input information.

Action:

Self-explanatory.

PS00-00

THE *xxxxxx* WAS SUCCESSFUL.
[*nnn* DATASET(S) DELETED]
['DEFAULTS' JOB USED]

Reason:

The requested function was successful. The *xxxxxx* indicates the function that was performed. If the DELete function was entered, *nnn* denotes the number of related data sets that were deleted. 'DEFAULTS' JOB USED is appended to the message when an ADD function was done and a DEFAULTS job was found in the database.

Action:

None.

PS00-01

A SERIOUS ERROR OCCURRED. ERROR CODES: WIP - *xx* RC - *yy*

Reason:

An error has occurred attempting to process the requested function.

Action:

Report this error, with the diagnostic information, to the CA Workload Automation SE technical staff at your site.

PS00-02

THE JOB WAS NOT FOUND.

Reason:

The requested job was not found on the CA Workload Automation SE database.

Action:

Check for correct input. If in error, correct and retry.

PS00-03

FUNCTION NOT SUCCESSFUL - DATABASE IS OUT OF SPACE.

Reason:

Part of the database is full, and the attempted function could not be completed successfully.

Action:

Report this error to the personnel responsible for CA Workload Automation SE maintenance at your site.

PS00-05

THE *xxxxxxxxxxxxx* DATA WAS INVALID OR OMITTED.

Reason:

The *xxxxxxxxxxxxx* indicates which field on the panel is causing the error.

Action:

Check for correct input. Ensure that all of the necessary fields were filled in with valid information. If JOBNAME, verify that first seven characters of the job name match the CA Workload Automation SE log dump job name.

PS00-07

THE JOB ALREADY EXISTS.

Reason:

An ADD function was requested for a job that is already defined in the CA Workload Automation SE database.

Action:

Use the UPD function when changing an existing job.

PS00-08

YOU ARE NOT AUTHORIZED TO ACCESS THAT JOB.

Reason:

A function was requested for a job with a USERID (OWNER on the CPU Job Definition panel) or a UID that does not match yours.

Action:

Update the OWNER field on the CPU Job Definition panel with your USERID if you require access to the job.

PS00-09

YOU ARE NOT AUTHORIZED TO DO THAT FUNCTION.

Reason:

You do not have the level of authority required to enter the function selected.

Action:

Contact your security administrator if you should be able to enter the function.

PS00-11

THE JOB IS NOT IN THE JCL LIBRARY SPECIFIED.

Reason:

The JCL member was not in the JCL library referenced.

Action:

Verify the location of the JCL and enter the correct JCL index. A /DISPLAY,ST=JCL topline command in CA Workload Automation SE can show the JCL data sets and the index number associated with them.

PS00-12

THE SPECIFIED PATTERN IS NOT FOUND.

Reason:

The PATTERN that was entered is not defined to CA Workload Automation SE.

Action:

Perform the function with a valid PATTERN name.

PS00-13

THE PATTERN SCHEDULE IS INVALID.

Reason:

The PATTERN that was entered is either not RESOLVed or has more than two SCHIDs associated with it.

Action:

Perform the function with a valid PATTERN name.

PS00-15

PATTERN AND TIME MUST BOTH BE SPECIFIED.

Reason:

These fields are both required when either one is entered.

Action:

Perform the function again with valid data in both PATTERN and TIME fields.

PS00-16

REQUEST REJECTED BY SASSXX14 USER EXIT

Reason:

Your installation user exit has rejected the request. The line following this message contains a message from the installation user exit indicating why the request was rejected.

Action:

Correct the problems indicated and reissue the request.

PS00-20

JOB IS NOT DEFINED AND NO JCL ID OR JCL LIB WAS SUPPLIED.

Reason:

A SUBMIT function was entered for a job that is not defined to CA Workload Automation SE and no JCL ID or JCL LIB was specified.

Action:

Enter the JCL ID or the JCL LIB if the job is not defined in the CA Workload Automation SE database.

PS00-21

THE JOB CAN NOT BE SUBMITTED.

Reason:

A SUBMIT function was entered for a job that cannot be brought into the CA Workload Automation SE request queue.

Action:

Contact your CA Workload Automation SE administrator, who may try a DEMAND command to produce other messages.

PS00-22

CAN NOT RETRIEVE THE JCL FOR THE JOB.

Reason:

A SUBMIT function was entered for a job and there was a problem accessing the JCL.

Action:

Contact your CA Workload Automation SE administrator, who may try a DEMAND command to produce other messages.

PS00-23

CAN NOT RETRIEVE THE JCL FROM THE OVERRIDE LIBRARY.

Reason:

A SUBMIT function was entered for a job that is set up to execute from the override library and there was a problem submitting that JCL.

Action:

Verify that the member exists in the override library (index 254). If the JCL should not be in the override library, update the CPU JOB DEFINITION panel and change the USE-OVRD LIBRARY from Y to N and resubmit. You may need to contact your CA Workload Automation SE administrator to do a DEMAND command to produce different diagnostic messages.

PS00-24

A JOB IS ALREADY SCHEDULED WITH THE SAME DUE OUT TIME.

Reason:

A SUBMIT function was entered for a job while the same job is in the CA Workload Automation SE queues with the same due-out date and time.

Action:

Wait a few minutes and perform the SUBMIT function again if you want duplicate job entries in the CA Workload Automation SE queues.

PS00-25

AN ERROR OCCURRED RETRIEVING THE SCHEDULE FOR THE JOB.

Reason:

The PATTERN job's schedule is in error.

Action:

Contact your CA Workload Automation SE administrator to determine the problem with the PATTERN job's schedule.

PS00-26

CONFLICT IN JCLID AND JCLLIB FIELDS

Reason:

The JCL library for a job can only be specified by a JCLID or a JCLLIB, not both.

Action:

Specify either a JCLID or a JCLLIB and reenter.

PS00-30

***xxxxxxxx* is an *yyyyy* JOB- NO *yyyyy* ENVIRONMENT EXISTS**

Reason:

xxxxxxxx is the job name. CA Workload Automation SE has been started with the cross-platform job feature disabled where *yyyyy* denotes the cross-platform job type (Agent or XPJOB).

Note: For more information about the XPDEF statement, see the *Systems Programming Guide*.

Action:

Enable the cross-platform job feature. Contact your installation specialist for assistance.

PS00-31

xxxxxxxx* FAILURE GETTING *yyyyyyyy* INFO SASSJOBR RC=*xx

Reason:

CA Workload Automation SE was unable to access *yyyyyyyyy* (agent information for agent jobs or node/executable information for XPJOBs) for cross-platform job *xxxxxxxx*. As a result, cross-platform job *xxxxxxxx* is not executed.

Action:

Note the message and return code provided, and contact CA Support at http://ca.com/support for assistance.

PS00-32

xxxxxxxx* FAILURE GETTING *yyyyyyyy* INFO SASJE RC=*xx

Reason:

CA Workload Automation SE was unable to access *yyyyyyyyy* (agent information for agent jobs or node/executable information for XPJOBs) for cross-platform job *xxxxxxxx*. As a result, cross-platform job *xxxxxxxx* is not executed.

Action:

Note the message and return code provided, and contact CA Support at http://ca.com/support for assistance.

PS00-33

***xxxxxxxx* FAILURE ADDING *yyyyyyyy* SEGMENT EOF REACHED**

Reason:

The *yyyyyyyyy* segment (agent information for agent jobs or node/executable information for XPJOBs) for cross-platform job *xxxxxxx* could not be added to the queue because it is full.

Action:

Perform one the following actions:

- Enlarge the queue if the queue if full.
- Post existing jobs before processing additional requests.

Contact your installation specialist for assistance.

PS00-34

xxxxxxxx* FAILURE ADDING *yyyyyyyy* SEGMENT SQADD RC=*xx

Reason:

The *yyyyyyyyy* segment (agent information for agent jobs or node/executable information for XPJOBs) for cross-platform job *xxxxxxx* could not be added to the queue.

Action:

Note the message and return code provided, and contact CA Support at http://ca.com/support for assistance.

PS00-99

AN INVALID MESSAGE NUMBER WAS SPECIFIED.

Reason:

An undefined message index number has been produced.

Action:

Contact your CA Workload Automation SE administrator who may need to contact CA Support.

Index

A

abends • 89
access level • 482
access restrictions • 21
accessing CA-7/PS • 477
accessing files • 311
active area
  appending JCL card-image data • 306
  defined • 335, 336
  saving documentation • 254
  saving schedule IDs • 95
Active job resources display panel • 190
ADD command • 474, 479
ADD function, alias • 34
adding
  application data sets at new site • 409
  CA Librarian JCL at new site • 413
  CA Panvalet for z/OS JCL at new site • 416
  cataloged procedures at new site • 419
  CPU job • 40
  data set • 151, 266, 297, 299
  DD level documentation member to database • 270
  documentation at new site • 421
  input network schedule to database • 108
  input network SCHID to work area • 111
  job at new site • 400
  job level documentation member to database • 255
  job SCHID to work area • 95
  job to database • 41
  network at new site • 407
  network level documentation member to database • 258
  network to database • 291, 292
  output network • 120, 123
  PDS • 306, 411
  schedules at new site • 423
  system level documentation member to database • 274
  user level documentation member to database • 262
address space resources • 174
adjusting schedule days • 95, 111
Agent DIV file • 428
Agent Job Definition panel • 71
agent jobs • 39
AL2PSBTI member • 482
aliases • 34, 37
alternate JCL libraries • 333
analyze
  application • 393
  commands • 384
APA reporting • 25
APPEND function, alias • 34
appending
  data set level documentation data • 266
  DD level documentation data • 270
  documentation data • 274
  JCL card-image data in active area • 306
  job level documentation data • 255
  network level documentation data • 258
  user level documentation data • 262
application data sets • 409
Application System Documentation panel • 274
AR.3 panel • 218
ARF (Automated Recovery Facility)
  AR.3 panel • 218
  ARF Condition Definition Edit panel • 220
  ARF statement • 319
  overview • 20, 209
ARF statement • 318
ARFSETs
  defining • 210
  overriding • 319
Automated Performance Analysis (APA) reporting • 25
availability • 26

B

base calendars
  and database transportability • 390
  defined • 83, 84
Batch Card Load Program (BCLP) • 23
batch input • 35
batch terminal • 482
BTI
  add CPU job to database • 40
  add data set to database • 297

add network to database • 291
analyze commands • 384
change network characteristics • 295
changing jobscharacteristics • 57
creating documentation commands • 254
bypassing
menus/panels • 31, 36
normal scheduling activities • 90
primary menus • 28

C

CA CPM • 27
CA Dispatch and demand networks • 123
CA Driver • 22
CA Earl reporting • 25
CA Easytrieve reporting • 25
CA JCLCheck Workload Automation • 330
CA Librarian JCL
database transportability • 412
maintaining • 306
Member Adds Report (SASSDT30-06) • 414
Member Deletes Report (SASSDT30-07) • 415
reviewing • 306
CA Panvalet for z/OS JCL
database transportability • 415
maintaining • 306
Member • 417, 418
Adds Report (SASSDT30-08) • 417
Deletes Report (SASSDT30-09) • 418
CA WA Restart Option
considerations • 21
step insertion • 390
CA-7 Function Menu • 477
CA7RSRC • 444
calendar • 482
Calendar macro • 84
calendar schedules • 390
calendar-based schedule • 291
cataloged procedures • 392, 418
changing
alias • 34
data set • 299, 304
DCB attribute • 304
JCL • 57
job • 41, 57
network characteristics • 295
panels • 33
requirements • 146
schedule • 87
CLEAR function, alias • 34
clearing panel
of all input data • 41
of input data and reset ID-COUNT • 262
command data sets • 435
commands
ADD • 474
DEL • 474
DEMAND • 90
DEMANDH • 90
DMDNW • 90, 291
DSN • 299
JCLxx • 316
JOB • 41
JOBCONN • 150
JOBCONN,DSN • 151
JOBCONN,JOB • 155
JOBCONN,NWK • 160
JOBCONN,RPT • 170
JOBCONN,USR • 166
LIST • 288, 474
LJCK • 316
LOAD • 331
LOADH • 331
LPROS • 253, 274, 288
NETWORK • 292
PRINT • 84
PROS,DD • 270
PROS,DSN • 266
PROS,JOB • 255
PROS,NWK • 258
PROS,USER • 262
PROSE • 254
PROSE,SYS • 274
PS • 477
QUIT • 474
RESOLV • 86
RUN • 90
RUNH • 90
RUNNW • 91
SCHD,DTRG • 131
SCHD,INWK • 108
SCHD,JTRG • 128
SCHD,ONWK • 120
STATUS • 474
SUBMIT • 474
UPD • 474
commas in data • 393

common formatted panel fields • 29
condition code tests • 322
conditions for adding job • 40
control statements
  #ARF • 319
  #JEND • 315
  #JI • 312
  #JO • 312
  #MSG • 320
  #RES • 321
  #SCC • 322
  #XEND • 316
  #XI • 316
  #XO • 316
control statements, reserved • 311
corequisite resources • 174
Corequisite resources list panel • 198
CPM • 186
CPU
  Job Definition panel • 41
  job dependency relationships • 57
  Job Documentation panel • 255
  Job Scheduling Parameter Edit commands • 95
  Job Scheduling Parameter Edit panel • 95
creating
  command files • 429
  processing schedules • 86
  temporary JCL • 332, 333
criteria for scheduling • 95
Critical Path Monitoring • 186

D

Data Set
  Definition panel • 299
  Documentation panel • 266
  Predecessors panel • 151
  Triggering panel • 131
data sets
  adding • 297
  allocation • 21
  connections • 155
  defined as preexecution requirement • 299
  dynamically allocated • 298
  level documentation • 266
  maintaining • 297
  triggers • 89, 424
database
  Adds Part 1 report (SASSDT30-01) • 402
  Adds Part 2 report (SASSDT30-02) • 404
  Deletes report (SASSDT30-03) • 405
  Disables report (SASSDT30-04) • 406
  extraction • 395
  maintenance menu • 31
  resynchronizing with new JCL • 57
  transportability • 387, 438
  verification • 379
date and time stamp • 29
date/time-driven scheduling • 83, 84
DB
  command • 31
  commands • 28
  function, alias • 34
  menu panels • 28, 31
DB.1 panel
  adding CPU job to database • 40
  changing jobscharacteristics • 57
  entering CPU job data • 41
  JCL change made outside CA Workload Automation SE • 57
  job marked MAINT=Y • 155
DB.10 panel (XPJOBs) • 57
DB.11 panel (agent jobs) • 71
DB.2.2 panel
  defining calendar-based schedule • 291
  scheduling options for input networks • 108
DB.2.3 panel
  defining options for output network schedules • 120
  entering schedule parameters • 292
DB.2.4 panel
  defining job triggers • 88, 128
  reviewing job triggers • 128
DB.2.5 panel, defining network triggers • 88
DB.2.6 panel, defining data set triggers • 89
DB.2.7 panel
  changing existing schedule • 87
DB.3 Menu panel • 150
DB.3.1 panel
  adding data set requirements • 151
  defining data set as preexecution requirement • 299
  disconnecting old data set name • 304
  maintaining data set/job connections • 304
  modifying data set requirements • 151
DB.3.2 panel, establishing
  dependency relationships for CPU job • 57
  predecessor job requirements • 155

DB.3.3 panel • 291
DB.3.4 panel
connecting network to job • 123
defining job requirements for networks • 160
defining output network connection • 292
redefining job connection • 295
DB.3.6 panel • 166
DB.3.7 panel • 170
DB.4 Menu panel • 254
DB.4.1 panel • 255
DB.4.2 panel • 258
DB.4.3 panel • 262
DB.4.4 panel • 266
DB.4.5 panel • 270
DB.4.6 panel • 274
DB.5 panel
adding network to database • 291
changing network characteristics • 295
defining networks • 292
DB.6 panel
adding data set to database • 297
data set TYPE=PERM • 155
defining • 89, 298, 299
marking data set as permanent for all jobs • 151
updating • 299, 304
DB.7 panel • 57, 146, 306
DBMADDS1 report • 421
DD
function • 41
Statement Documentation panel • 270
deadline prompting • 41
DEFAULTS job record • 40, 56
defining
agent jobs • 71
alternate library • 333
annual schedule • 95, 111
calendar-based schedule • 291
connection • 150
daily schedule • 95, 111
data set level documentation • 266
data set triggers • 89
data sets that trigger job scheduling • 131
data sets with DB.6 panel • 299
database • 31
DD level documentation • 270
documentation • 253
dynamically allocated data set • 298
individual workstation schedules in output network • 123
input network connection • 291
job level documentation • 255
job requirements for networks • 160
job triggers • 88
jobs • 40
jobs that trigger scheduling of other jobs • 128
monthly schedule • 95, 111
network level documentation • 258
network triggers • 88
output network connection • 292
predecessor requirement • 150
requirements • 145
schedules for jobs run on date and time basis • 95
scheduling criteria • 95
scheduling options for input networks • 108
scheduling parameters for input networks scheduled on date/time basis • 111
special override library • 332
step level condition code tests • 322
symmetric schedule • 95, 111
system level documentation • 274
systems • 41, 57
triggers • 128
user level documentation • 262
weekly schedule • 95, 111
defining pattern jobs • 484
DEL command • 474
DELETE function, alias • 34
deleting
application data sets at original site • 410
CA Librarian JCL at original site • 414
cataloged procedures at original site • 419
data set and its elements from database • 299
data set level documentation member from database • 266
DD level documentation member from database • 270
documentation at original site • 422
documentation member from database • 274
input network schedule member from database • 108
input network SCHID from work area • 111
job and its data from database • 41
job and its data sets • 41
job at original site • 404
job level documentation member from database • 255
job SCHID from work area • 95

member of PDS containing JCL • 306
network and its documentation/schedules from database • 292
network at original site • 407
network level documentation member from database • 258
output network • 120, 123
PDS JCL members at original site • 412
schedules at original site • 426
user level documentation member from database • 262
DELPRRN function, deleting JCL in trailer queue • 41
DEMAND command
adding jobs • 40
and TYPE=RES • 155, 165
running jobs by request • 90
DEMANDH command • 90
dependence definition • 145
dependency relationships for CPU job • 57
device control, VRM • 203
diagnostic messages • 29
directly connected • 361
disabling
CA Panvalet for z/OS JCL at original site • 417
job at original site • 405
network at original site • 408
disconnecting old data set name • 304
discontinuing job • 41
displaying job status • 474, 481
DL parameter value • 312
DMDNW command
network definition requirements • 291
overview • 90
documentation
adding to database • 421
categories • 254, 277
defining segment names • 279
defining subsegment names • 282
displaying segments • 281
displaying subsegments • 284
END statement • 288
LINKed documentation • 394
naming conventions • 284
overview • 24
panels • 253
QDSEC statement • 286
RESTART statement • 288
segment format • 280
segment names • 278, 279
segmentation guidelines • 278
special features • 277
special purpose • 285
subsegment format • 283
subsegment guidelines • 282
user-defined segments • 278
workload • 253
DSN
command • 299
connections • 403
DSNBR (CA Workload Automation SE assigned) • 297
DT60ADDS file • 441, 444
due-out
day for network workstation • 111
time of day • 95, 111, 123, 131
dynamically allocated data set • 298

E

edit facility • 335
EDIT function
alias • 34
DB.2.2 panel • 108
DB.2.3 panel • 120
default requests • 363
edit panels
Input Network Scheduling Parameter Edit • 111
Job Scheduling Parameter Edit • 95
Output Network Scheduling Parameter Edit • 123
Edit Work File (EWF)
defined • 335
validating • 330
editing
data set level documentation • 266
DD level documentation • 270
JCL in PDS member • 306
job level documentation • 255
limitations • 367
network level documentation • 258
system level documentation • 274
user level documentation • 262
editor • 337
elapsed time for TRGD-JOB • 131
error messages • 484
event scheduling • 88
event-driven scheduling • 83
exception conditions and ARF • 210
exclusive resource • 174
execution

JCL libraries • 391
requirements • 145
EXIT function, returning to
DB.2.1 panel and restore data • 95
DB.2.2 and restore data • 111
DB.2.3 and restore data • 123
exiting
database maintenance mode • 28
from DB using batch input • 37
Expansion of Requested Jobs Report (SASSDT20-01) • 395
express priority • 41
External Communications Facilities • 23
external data set • 299
external security • 482

F

FE function
alias • 34
FEM (Full Edit Mode) • 340
command descriptions • 340
commands • 340
panel areas • 340
sample panels • 340
subcommands • 340
FETCH and EDIT
data set level documentation • 266
DD level documentation • 270
input network schedule data • 108
JCL in PDS member or sequential data set • 306
job level documentation • 255
network level documentation • 258
output network schedule • 120
system level documentation • 274
user level documentation • 262
FETCH function
alias • 34
FETCH function, retrieving
data set level documentation • 266
DD level documentation • 270
input network schedule data from database • 108
JCL data • 306
job level documentation • 255
network level documentation • 258
output network data from database • 120
system level documentation • 274
user level documentation • 262
FETCHP function, alias • 34
fields common to all formatted panels • 29
flows • 186
FORMAT function
alias • 34
DB.1 panel • 41
formatted panels
common fields • 29
functions/alias names • 34
using for online input • 28
function
ADD • 41, 474
alias names • 34
APPEND • 255
DD • 41
DEL • 474
DELETE • 41
DELPRRN • 41
field • 28, 474
FORMAT • 41
LIST • 41, 474
Menu panel • 30
PS • 477
PURGE • 41
QUIT • 474
RENAME • 299
RUN • 306
RUNH • 306
shortcuts • 33
STATUS • 474
SUBMIT • 474
transfer • 33, 37
UPD • 41, 474

G

GDG • 299

H

Health Checker feature • 26
HELP facility • 17
HELP library • 41
history reporting • 25
HLD statement • 318
HOLD field • 41

I

IEHPROGM-type functions • 21
Implementing CA-7/PS • 482, 484

including a job • 312
initialization file
  and JCL library • 306
  DAIO statement • 336
  date and time scheduling parameters • 84
  DEFAULTJOB parameter • 56
  schedule scan parameters • 88
  STATIONS statement • 288
Input Network
  database transportability • 425, 426
  defining • 291, 292
  Scheduling panel • 108
  scheduling parameter edit commands • 111
  Scheduling Parameter Edit Panel • 111
input panels • 28
input workstation networks • 408
Input/Output Network
  Definition panel • 292
  Documentation panel • 258
  Tasks panel • 160
in-stream JCL procedures • 392
interfaces
  to external security packages • 21
  TSO/ISPF • 361
internal security • 482
ISPF • 361, 365

J

J statements • 311
JCL
  changes • 57, 146
  command • 306
  libraries • 306, 328, 330, 333
  Library Maintenance panel • 306
  management • 306
  overrides • 22, 311
  panels • 306
  special override library • 331
  statement • 318
  statements (using DB.7 panel) • 306
  syntax checking • 330
  temporary • 332, 333
  utility • 328
  validation • 330
JCL ID field • 474
JCL LIB field • 474
JCLCHECK • 330
JCLID field • 41
JCLOVRD command • 41
JCL-OVRD field • 41
JCLxx command • 316
JEND statement • 315
JI statement • 312
JO statement • 312
job
  class • 41
  command • 41
  connections • 402
  defining agent jobs • 71
  defining CPU jobs • 41
  defining XPJOB jobs • 57
  Definition panel • 41
  dependency, predecessor job requirements • 155
  discontinuation of • 41
  field • 41
  include • 312
  level documentation • 255
  Mass Job Rename utility • 453
  message, routing to logical terminal • 41
  name UCC7Rxxx, for workload balancing • 41
  omit • 312
  panel • 41
  predecessor/dependency relationship • 291
  Predecessor/Successor Menu • 150
  priority • 41
  requirements • 155
  resource management panel • 180
  restart • 21
  Scheduling Parameter Edit commands • 95
  Scheduling Parameter Edit panel • 95
  timing, start of • 41
  Triggering panel • 128
  triggers • 88, 423
JOB DOES NOT CURRENTLY HAVE A SCHEDULE status message • 474
JOB IS ACTIVE ON CPU status message • 474
JOB IS AWAITING EXECUTION status message • 474
JOB IS NEXT SCHEDULED AT hhmm ON yy.ddd status message • 474
JOB IS WAITING FOR JOB xxxxxxxx status message • 474
JOB LAST RAN AT hhmm ON yy.ddd status message • 474
Job Predecessors panel • 155
JOB RUNS AFTER field • 474
job/network connections • 295

Job/resource cross reference list panel • 187
JOBCONN command • 150
JOBCONN,DSN command • 151
JOBCONN,JOB command • 155
JOBCONN,NWK command • 160
JOBCONN,RPT command • 170
JOBCONN,USR command • 166
Jobflow Monitor • 27
  overview • 27
JOBNAME field • 474
JOBREPT file • 444
jobs
  agent • 39
  CPU • 39
  defining • 40
  dependency • 145
  documentation • 253
  Mass Job Rename utility • 453
  panel • 41
  predecessor • 145
  Requested To Be Moved Report (SASSDT10-01) • 395
  scheduling • 83
  separating • 155
  successor • 145
  waiting on resources panel • 196
  XPJOB • 39

L

lead time
  for schedule ID • 95, 123
  for TRGD-JOB • 131
  for workstation • 111
LINKed documentation • 394
list
  function alias • 34
  function shortcuts • 33
LIST command • 288, 474
listing
  base calendars • 84
  data set level documentation • 266
  DD level documentation • 270
  job names and schedule IDs for triggering element • 131
  SCHIDs for • 95, 111, 123
    input network • 111
    job • 95
    output network • 123
  system level documentation • 274
LJCK command • 316
LNTWK command • 295, 407
  and database transportability • 407
  using when changing a network • 295
LOAD command
  adding jobs • 40
  changing jobs • 57
  creating job profile data • 331
load process
  adding data sets • 297
  and changing JCL • 57
  and dynamically allocated data sets • 298
  and execution requirements • 145
LOAD/RELOAD status • 389
LOADH command • 41, 331
Logon Menu • 477
LPROS command loop
  DB.4.3 panel • 262
  DB.4.4 panel • 266
  DB.4.5 panel • 270
  DB.4.6 panel • 274
LPROS command with #RESTART • 288

M

maintaining data set/job connections • 304
management level reporting • 24
Mass Job Rename utility • 453
master station • 320
menu
  bypassing secondary • 32
  DB (database maintenance) • 31
  documentation • 254
  function • 30
  Job Predecessor/Successor • 150
  RM (resource management) • 179
  VRM • 179
menu (CA-7/PS) • 477
MESSAGE • 29, 41
  field • 29
  routing to logical terminal • 41
messages • 484
MIXED FEM subcommand • 340
MNT statement • 318
modifying
  alias • 34
  data set requirements • 151
  processing schedules • 86

schedule • 87
monitoring • 27
workload • 27
month-by-month listing of base calendars • 84
MSG statement • 318, 320
MSG-INDX field • 29
MSGS field • 29
mutual exclusion • 155

N

naming conventions for documentation • 284
network
and job connections • 295
command • 292
connections • 291
defining • 291
deleting • 292
demanding • 90
documentation • 254
level documentation • 258
overview • 291
panel • 291
triggers • 88, 424
NORM-type data set • 299
NOX statement • 318
NTR statement • 318
NWK connections • 402
NXTCYC settings • 391

O

omit job • 312
online
assistance • 17
documentation • 253
input • 28
utilities • 21
on-request scheduling • 40, 89
Output Network
commands when moving • 426
defining • 292
schedule • 120, 426
Scheduling panel • 120
scheduling parameter edit commands • 123
Scheduling Parameter Edit panel • 123
output workstation network • 408
overriding • 311, 319
ARFSETs • 319
JCL • 311
OWNER field • 474

P

PA key
assignment • 18
using with ISPF editor • 365
using with text editor • 360
panel access • 482
panel ID • 482
panel ID field • 29
Panel to panel transfer • 33
panels
Active Job Resources Display • 190
Agent Job Definition • 71
Application System Documentation • 274
ARF Condition Definition Edit • 220
ARF Condition Definition Maintenance • 218
bypassing • 31
common fields • 29
Corequisite Resources list • 198
CPU Job Definition • 41
CPU Job Documentation • 255
Data Set Definition • 299
Data Set Documentation • 266
Data Set Predecessors • 151
Data Set Triggering • 131
DB Menu • 31
DD Statement Documentation • 270
edit • 123
Function Menu • 30
Input Network Scheduling • 108
Input Network Scheduling Parameter Edit • 111
Input/Output Network Definition • 292
Input/Output Network Documentation • 258
Input/Output Network Tasks • 160
JCL Library Maintenance • 306
Job Predecessor/Successor Menu • 150
Job Predecessors • 155
Job Resource Management • 180
Job Scheduling Parameter Edit • 95
Job Triggering • 128
Job/resource Cross Reference List • 187
Jobs Waiting on Resources • 196
layout • 29
Output Network Scheduling • 120
Output Network Scheduling Parameter Edit • 123
Pending Resources Job Display • 193
Report IDs Created • 170

Resource Count Resource Management • 200
RM menu • 179
User Memo-Form Predecessors • 166
User-Defined Item Documentation • 262
Workload Documentation Menu • 254
XP Job Definition • 57
PATTERN field • 474
PATTERN jobs • 482, 484
PDS JCL members • 410
Pending resources job display panel • 193
PERM data set • 298, 299
permanent data sets • 298
perpetual calendars • 85
Personal Scheduling • 473
PF key
assignment • 18
using with ISPF editor • 365
using with text editor • 360
pointer verification • 379
POST AT CLOSE TIME field • 89, 299
POST command • 148, 166
predecessor requirements
defining • 155
dependency • 145, 155
satisfying • 148
temporary • 147
predecessor/successor relationships • 402
preexecution requirement • 41, 299
PRINT command • 84
printing contents of JCL members in PDS • 328
PROCLIB Member Deletes Report (SASSDT30-05) • 421
producing month-by-month listing of base calendars • 84
productivity • 24
PROGRAM field • 29
prompting messages, issuing if job late • 41
PROMPTS field • 41
PROS,DD command • 270
PROS,DSN command • 266
PROS,JOB command • 255
PROS,NWK command • 258
PROS,USER command • 262
PROSE command • 254
PROSE,SYS command • 274
PS command • 477
PS00 application • 482

Q

QM.1 panel
and posting requirements • 148
indicating task completion • 166
QM.2 panel
indicating task completion • 166
viewing requirements of a job • 148
QUIT command • 474
QUIT function, alias • 34

R

RDAY field • 95, 111
redefining job connection • 295
relative days • 95, 111
removing
expired CA Workload Automation SE JCL Scheduled Override control statements • 328
network station • 295
scheduled JCL • 328
RENAME
function • 299
function, alias • 34
RENAMES control file • 454
renaming
data set • 299, 304
Mass Job Rename utility • 453
PDS member containing JCL • 306
REPL function, alias • 34
replacing
data set level documentation member • 266
DD level documentation member • 270
input network • 108, 111
schedule in database • 108
SCHID in work area • 111
job • 255
level documentation member • 255
network level documentation member • 258
output network SCHID in work area • 123
PDS member or sequential data set containing JCL • 306
schedule member in database • 120
SCHID in work area • 95
system level documentation member • 274
user level documentation member • 262
Report IDs Created panel • 170
reporting
Automated Performance Analysis • 25
CA Earl reporting • 25

CA Easytrieve reporting • 25
CA Panvalet JCL Member Adds Report (SASSDT30-08) • 417
CA Panvalet JCL Member Deletes Report (SASSDT30-09) • 418
Database Adds - Part 1 Report (SASSDT30-01) • 402
Database Adds - Part 2 Report (SASSDT30-02) • 404
Database Deletes Report (SASSDT30-03) • 405
Database Disables Report (SASSDT30-04) • 406
Expansion of Requested Jobs Report (SASSDT20-01) • 395
history • 25
Jobs Requested To Be Moved Report (SASSDT10-01) • 395
LIBRARIAN JCL Member Adds Report (SASSDT30-06) • 414
LIBRARIAN JCL Member Deletes Report (SASSDT30-07) • 415
PROCLIB Member Deletes Report (SASSDT30-05) • 421
productivity • 24
requirements
automatically tracked and satisfied by system • 148
definitions • 145
needing manual intervention • 148
network connections • 402
posting • 299
satisfied • 166
RES statement • 318, 321
RESANL command • 41, 384
Reserved DDname Table • 298
RESOLV • 86, 108
command • 86
function • 86, 108
overview • 86
using with DB.2.2 panel • 108
RESOLV function alias • 34
Resource count resource management panel • 200
resource count resources • 175
resource name • 482
resources
address space • 174
corequisite • 174
exclusive • 174
handling • 179
resource count • 175
shared • 174
VRM variable definitions • 175
restricting access • 21
resynchronizing database with new JCL • 57
retrieving
data set level documentation • 266
DD level documentation • 270
input network schedule data from database • 108
JCL data • 306
job • 255
level documentation • 255
network level documentation • 258
output network data from database • 120
system level documentation • 274
user level documentation • 262
returning
to DB.2.1 panel and restoring data • 95
to DB.2.2 panel and restoring data • 111
to DB.2.3 panel and restoring data • 123
reviewing
agent job data • 71
CA Librarian JCL • 306
CA Panvalet for z/OS JCL • 306
CPU job data • 41
data sets that trigger job scheduling • 131
individual workstation schedules in output network • 123
JCL statements • 306
jobs that trigger scheduling of other jobs • 128
schedules for jobs run on date and time basis • 95
scheduling options for input networks • 108
scheduling parameters for input networks scheduled on date and time basis • 111
XPJOB job data • 57
RM Menu panel • 179
RM.1 panel • 180
RM.2 panel • 187
RM.3 panel • 190
RM.4 panel • 193
RM.5 panel • 196
RM.6 panel • 198
RM.7 panel • 200
ROLL field
defining/reviewing schedules • 95
defining/reviewing schedules for input networks • 111
rolling schedule forward/back • 95, 111

routing message about job to logical terminal • 41
RQMT command • 384
RQVER command • 384
RSRC keyword • 186, 203
RUN command
  and true predecessor requirements • 155, 165
  bypassing normal scheduling • 90
RUN function • 306
RUNH command
  assigned to class 9 • 41
  bypassing normal scheduling • 90
RUNH function on DB.7 panel • 306
RUNNW command • 91, 291

S

SASSBSTR program • 432
SASSDT10
  execution JCL • 432
  module • 432
SASSDT10-01 report • 395, 431
SASSDT20 execution JCL • 433
SASSDT20 module
  CA Librarian JCL • 412
  CA Panvalet for z/OS JCL • 415
  cataloged procedures • 418
  expanding requested job • 395
  PDS JCL members • 410
SASSDT20-01 report • 395, 433
SASSDT30
  execution JCL • 438
  module • 395, 434, 435
SASSDT30-01 report
  adding at new site • 407, 410, 411, 421
    application data sets • 410
    documentation • 421
    PDS JCL members • 411
    workstation networks • 407
  sample report • 402
SASSDT30-02 report
  adding schedules at new site • 423, 425, 426
  sample report • 404
SASSDT30-03 report • 404, 405, 422
SASSDT30-04 report • 406
SASSDT30-05 report • 419, 421
SASSDT30-06 report • 414
SASSDT30-07 report • 415
SASSDT30-08 report • 417
SASSDT30-09 report • 418
SASSDT30-10 report • 427
SASSDT30-11 report • 428
SASSDT60
  execution JCL • 442
  module • 441
SASSDT60-01 report • 441
SASSDTAB table • 203
SASSJCLU batch utility • 328
SASSMSGS module • 286, 287
SASSPMDD table • 298
SASSUTBL module • 41
satisfying requirements • 148
SAVE function
  alias • 34
  replacing sequential data set containing JCL • 306
SAVE function, adding
  data set level documentation member to database • 266
  DD level documentation member to database • 270
  input network schedule to database • 108
  job level documentation member to database • 255
  job SCHID-COUNT and SCHID in work area • 95
  network level documentation member to database • 258
  output network schedule to database • 120
  PDS member containing JCL • 306
  system level documentation member to database • 274
  user level documentation member to database • 262
SAVE function, updating
  input network SCHID-COUNT and SCHID in work area • 111
  output network SCHID-COUNT and SCHID in work area • 123
SCAL values • 482
SCC statement • 318, 322
SCHD,DTRG command • 131
SCHD,INWK command • 108
SCHD,JTRG command • 128
SCHD,ONWK command • 120
schedule
  adding at new site • 423
  defining • 91
  deleting at original site • 426
  ID, importance of • 83
  modifications • 87, 390

resolving • 86
triggers • 88, 89
data set • 89
job • 88
network • 88
schedule scan
date and time scheduling • 84
overview • 88
scheduling
entering parameters • 292
environment • 175
JCL overrides • 311
job, trigger • 128
overview • 83
relative days • 95, 111
SCHENV keyword • 175, 180
SCHID field • 95, 111, 123, 131
security
external • 482
internal • 482
module • 482
selecting
database maintenance function • 31
documentation functions • 254
Sending messages • 320
sequence numbers • 394
shared resources • 174
shortcuts
bypass menus/panels • 31, 36
LIST function • 33
SMF interface • 299
special
documentation features • 277
documentation messages • 292
override library • 41, 331
purpose documentation • 285
SR (SAVE/REPL) function
input network schedule • 111
job schedule • 95
output network schedule • 123
SS (SAVE/SAVE) function
SAVE input network schedule in database • 111
SAVE job schedule in database • 95
SAVE output network schedule in database • 123
STATUS command • 474, 481
STATUS function, alias • 34
status messages • 474
SUBMIT command • 474, 480
submitting
JCL in text editor active area to default MAINID • 306
time of day • 95, 131
SUBTM command • 148
SYSIN control cards • 443
System level documentation panel • 274
System Management Facility (SMF) interface • 299

T

task completed • 166
temporary JCL • 332, 333
temporary predecessors • 147
tertiary menus • 32
text editor • 25, 335
TIME field • 474
timing
for generation of base calendar • 84
permanent discontinuation of job • 41
start of new jobs • 41
top line commands • 28
transferring
from panel to panel • 33
to another DB function (batch input) • 37
to Input Network Scheduling Parameter Edit panel • 108
to Output Network Scheduling Parameter Edit panel • 120
work from one database to another • 387
TRIG command • 384
trigger scheduling of job • 128
triggers • 88, 423, 424
TSO/ISPF interface • 361
TYPE field • 299

U

UCC7Rxxx, job name for workload balancing • 41
UID • 474
unconnected workstation networks • 408
unsatisfied requirements • 148
UPD command • 474
UPD function, alias • 34
updating
data set characteristics • 304
data set information in database • 299
data set requirements • 151
DESC and LINK fields • 255, 274
input network SCHID-COUNT and SCHID in work area • 111

job characteristics • 57
job information in database • 41
job SCHID-COUNT and SCHID in work area • 95
network information in database • 292
network requirements • 160
output network SCHID-COUNT and SCHID in work area • 123
predecessor job requirements in database • 155
to remove network station • 295
triggering element • 131
user requirement • 166
UPPER FEM subcommand • 340
user
ID security • 394
level documentation • 262, 392
Memo-Form Predecessors panel • 166
requirement • 166
User-defined
documentation segments • 278
Item Documentation panel • 262
USERIDs • 482
using temporary JCL • 332, 334
USR connections • 403
Utility commands • 21, 299, 304
RENAME • 299, 304

V

Variable parameter in ARF • 248
VER statement • 318
VERIFY command • 41, 148
Verifying logical pointers in database • 379
VRM (Virtual Resource Management)
and database transportability • 427, 440
and mass job renaming • 453
DBT job extract JCL • 442
device control • 203
Job Extract report • 441
Menu panel • 179
overview • 173
variable definitions • 175
VRMXREFS data set • 427

W

WLB • 24, 41
WLM • 175, 180
WLM (IBM Workload Manager) • 19
work flow control
commands
DEMAND • 40, 90, 155, 165
DEMANDH • 90
DMDNW • 90, 291
JCLOVRD • 41
LOAD • 40, 57, 155, 165, 331
LOADH • 41, 331
POST • 148, 166
RUN • 40, 90, 155, 165
RUNH • 41, 90
RUNNW • 91, 291
SUBTM • 148
VERIFY • 41, 148
work scheduling • 83
workload
documentation • 24, 253
documentation menu panel • 254
forecasting • 23
management • 175
planning • 24
scheduling • 18
sequencing • 18
workstation networks • 291, 392, 406

X

XEND statement • 316
XI statement • 316
XN.1 panel • 367
XN.2 panel • 370
XNODE command • 367
XO statement • 316
XP Job Definition panel • 57
XPJOB jobs • 39
XPSWD command • 370
XREF command • 384