

CONTROL-M for z/OS

Getting Started Guide



Supporting

CONTROL-M for z/OS version 6.2.18

July 31, 2006

Contacting BMC Software

You can access the BMC Software Web site at <http://www.bmc.com>. From this Web site, you can obtain information about the company, its products, corporate offices, special events, and career opportunities.

United States and Canada

Address BMC Software, Inc.
2101 CityWest Blvd.
Houston TX 77042-2827

Telephone 713 918 8800 or
800 841 2031

Fax 713 918 8000

Outside United States and Canada

Telephone (01) 713 918 8800

Fax (01) 713 918 8000

Copyright 2006 BMC Software, Inc., as an unpublished work. All rights reserved.

BMC Software, the BMC Software logos, and all other BMC Software product or service names are registered trademarks or trademarks of BMC Software, Inc.

All other trademarks belong to their respective companies.

BMC Software considers information included in this documentation to be proprietary and confidential. Your use of this information is subject to the terms and conditions of the applicable End User License Agreement for the product and the proprietary and restricted rights notices included in this documentation.

Restricted Rights Legend

U.S. Government Restricted Rights to Computer Software. UNPUBLISHED -- RIGHTS RESERVED UNDER THE COPYRIGHT LAWS OF THE UNITED STATES. Use, duplication, or disclosure of any data and computer software by the U.S. Government is subject to restrictions, as applicable, set forth in FAR Section 52.227-14, DFARS 252.227-7013, DFARS 252.227-7014, DFARS 252.227-7015, and DFARS 252.227-7025, as amended from time to time. Contractor/Manufacturer is BMC Software, Inc., 2101 CityWest Blvd., Houston, TX 77042-2827, USA. Any contract notices should be sent to this address.

Customer Support

You can obtain technical support by using the Support page on the BMC Software Web site or by contacting Customer Support by telephone or e-mail. To expedite your inquiry, please see “Before Contacting BMC Software.”

Support Web Site

You can obtain technical support from BMC Software 24 hours a day, 7 days a week at http://www.bmc.com/support_home. From this Web site, you can

- read overviews about support services and programs that BMC Software offers
- find the most current information about BMC Software products
- search a database for problems similar to yours and possible solutions
- order or download product documentation
- report a problem or ask a question
- subscribe to receive e-mail notices when new product versions are released
- find worldwide BMC Software support center locations and contact information, including e-mail addresses, fax numbers, and telephone numbers

Support by Telephone or E-mail

In the United States and Canada, if you need technical support and do not have access to the Web, call 800 537 1813. Outside the United States and Canada, please contact your local support center for assistance. To find telephone and e-mail contact information for the BMC Software support center that services your location, refer to the Contact Customer Support section of the Support page on the BMC Software Web site at http://www.bmc.com/support_home.

Before Contacting BMC Software

Before you contact BMC Software, have the following information available so that Customer Support can begin working on your problem immediately:

- product information
 - product name
 - product version (release number)
 - license number and password (trial or permanent)
- operating system and environment information
 - machine type
 - operating system type, version, and service pack or other maintenance level such as PUT or PTF
 - system hardware configuration
 - serial numbers
 - related software (database, application, and communication) including type, version, and service pack or maintenance level
- sequence of events leading to the problem
- commands and options that you used
- messages received (and the time and date that you received them)
 - product error messages
 - messages from the operating system, such as file system full
 - messages from related software

Contents

About This Guide	15
<hr/>	
Format of This Guide	15
Before You Begin	17
Conventions Used in This Guide	17
Help Information	20
Information New to This Version	20
Related Publications	21
Chapter 1 Introduction to CONTROL-M	23
<hr/>	
Overview	24
Preparations	24
Main Components of CONTROL-M	24
Online Facility Primary Options	26
Introduction to the Scheduling Definition Facility	26
Introduction to the Active Environment Screen	36
Toggling Between Display of Commands and Options	36
Filtering the Active Environment Display	37
Confirming a Job Order	39
Checking the Log for the Job from the Active Environment Screen	41
Checking the Sysout for the Job	42
Holding and Deleting Active Environment Screen Job Orders	44
Return to the Scheduling Definition Facility	46
Saving Job Scheduling Definitions	46
Selecting a Table from the Table List Screen	48
Formats of the Job List Screen	50
Exiting the Online Facility	50
Review	51
Recommended Reading	52
Chapter 2 Scheduling Definition and Manual Intervention	55
<hr/>	
Overview	56
Preparations	56
Sections of the Job Scheduling Definition	57
General Parameters Section	59
Basis Scheduling Parameters Section	60
Runtime Scheduling Parameters Section	64
Post-Processing Parameters Section	67
Conditions, Date Qualifiers, and Job Dependencies	70

Displaying the Scheduling Plan for the Job	71
Defining JOB4	74
Job Ordering and Job Forcing	76
Why Screen	78
Why the Job Did Not Run	79
Adding a Missing Condition in the Why Screen	80
IOA Conditions/Resources Screen	81
Filtering the IOA Conditions/Resources Screen	81
Adding a Condition or Resource	82
A Second Look at the Why Screen	83
Zoom Screen	84
Freeing a Held Job	86
Review	88
Recommended Reading	89
Chapter 3 Restarts under CONTROL-M/Restart	91
Overview	92
Preparations	92
Defining Restart in the Job Scheduling Definition	93
Editing JCL from the Active Environment	99
Restarting the Job	100
Review	104
Recommended Reading	104
Chapter 4 Group Scheduling	105
Overview	106
Preparations	106
Creating a Group Scheduling Table	107
Group Entity	108
Planned Logic of the Job Scheduling Definitions	115
Jobs Scheduling Definitions in the Group Table	115
Saving the Table	126
Points to Remember Concerning Group Scheduling	126
Group Entity Ordering and Job Ordering	126
Advantages of Group Scheduling	130
Review	131
Recommended Reading	132
Chapter 5 AutoEdit and JCL	133
Overview	134
Preparations	135
Using AutoEdit System Variables in the JCL	135
System Date, Working Date and ODATE	137
Date System Variables	138
Non-Date System Variables	139
Introduction to AutoEdit Control Statements and Functions	140
AutoEdit Function Details	142

Additional Points About User Daily Jobs	201
Job Ordering Through Online Utility CTMJOBRO	203
Other Methods of Job Ordering	205
Review	206
Recommended Reading	206

Chapter 8 Additional Features **207**

Overview	208
Preparations	208
Defining Calendars	209
Structure of the IOA Calendar Definition Facility	214
Responding to External Events through CMEM	215
Event Types Handled by CMEM - Available ON Statements	221
Creating On Spool Jobs	222
Defining On Spool Jobs	223
Bringing the Job On Spool	227
Additional Points About On Spool Jobs	228
Batch Utility IOACND	229
Miscellaneous Facilities	232
History Jobs File	232
Journaling and Restoration Capability	232
Accumulating Statistics: Statistics Facility	233
Automatic Tape Adjustment	234
Simulating Job Execution and Forecasting Resource Usage	234
CONTROL-M/Restart Dataset Cleanup Utility CTRCCLN (R2)	235
Reporting Facility	235
Exit the Online Session	235
Review	236
Recommended Reading	237

Appendix A Sample JCLs **239**

Chapter 1	239
JOB1	239
JOB2	239
Chapter 2	240
JOB3	240
JOB4	240
Chapter 3	241
JOB5	241
Chapter 4	242
JOB1	242
JOB2	242
JOB3	242
JOB4	243
JOB5	244
Chapter 5	245
JOB6	245
JOB7	245

JOB8.....	245
JOB9.....	245
Chapter 6.....	246
JOB3.....	246
JOB5.....	247
Chapter 7.....	247
Chapter 8.....	248
JOB10.....	248
EXT1.....	248

Index	249
--------------	------------

Figures

CONTROL-M Scheduling Definition Facility - Entry Panel	26
CONTROL-M Scheduling Definition Facility - Entry Panel	27
Job Scheduling Definition Screen	28
Job Scheduling Definition Screen Showing Completed Fields	30
Job List Screen	31
Job List Screen	34
Confirm Window	35
Job Order Messages Screen	35
CONTROL-M Active Environment Screen	36
Show Screen Filter Window	38
Filtered Display in the Active Environment Screen	39
Confirmation Window	40
Log Option in the Active Environment Screen	41
CONTROL-M Log Screen	41
Job Order Execution History Screen	43
Sysout Viewing Screen	44
Delete Option in the Active Environment Screen	44
Job List Screen Exit Option Window	47
CONTROL-M Table List Screen	48
Job List Screen	49
CONTROL-M Scheduling Definition Facility - Entry Panel	57
Job Scheduling Definition Screen	58
Option P (PLN)	72
Job Scheduling Plan Window	72
Job Scheduling Screen	73
Job Scheduling Definition for job IDJOB4	75
Job List Screen	76
Job List Screen	77
Why Screen	79
Confirmation Window for Adding Condition in Why Screen	80
IOA Conditions/Resources Screen	81
Add Resources Window in IOA Conditions/Resources Screen	82
Zoom Screen	85
Job Scheduling Definition	97
Confirm Restart Window	100
Restart Step List Window	102
Group Entity Scheduling Definition	108
Basic Scheduling Parameter Section	111
Post-Processing Parameter Section	113
Runtime Scheduling Parameter Section	117

Job Scheduling Definition for IDJOB4	123
Job List Screen for a Group Entity	126
Sysout for Job IDJOB6	136
AutoEdit Simulation Panel	148
AutoEdit Simulation Output	152
Show Screen Filter Window	171
Active Environment Screen Display Filters Window	174
Display Type A (All Fields)	176
Sysout Viewing Screen	184
Group Screen	189
IOA Calendar Definition Facility - Entry Panel	210
IOA Calendar Definition Screen	211
Year List Screen	213
CMEM Rule Definition Facility - Entry Panel	215
CMEM Rule Definition Screen	217
CMEM Rule List Screen	219
CMEM Exit Option Window	220
CONTROL-M CMEM Order Messages Screen	220

Tables

Job Scheduling Definition Screen Field Descriptions	28
Values for Job Scheduling Definition Screen Fields	29
Commands to Display Job List Screen Formats	50
Job Scheduling Definition Screen General Parameters	60
Basic Job Scheduling Definition Parameters	61
Runtime Scheduling Parameters	64
Post-Processing Parameters ON and DO Statements	67
PF Keys and Screen Navigation Commands for Job Scheduling Plan Screens	73
DO IFRERUN: Selected FROM Keyword values	96
Valid Values For the ON GROUP-END Statement	112
Non-Date AutoEdit System Variables	140
Other AutoEdit Functions	144
Pointing to External Members	153
Additional AutoEdit Control Statements	154
Valid AutoEdit Operators	156
IDJOB8 SET VAR Statement Owners	160
Anticipated Results of Job	161
Job Dependency Fields	181
Job Information	183
Job Execution Information	183
Format of the Second Date Control Record (For Enhanced Daily Checkpointing Only)	200
Non-Routine Job Ordering Methods	205
ON Statements Available to CMEM	221
CMEM Action Parameters	222
Job To Be Brought On Spool - Values for Exercise	223
CMEM Rule - Values for Exercise	223
Job Scheduling Definition - Values for Exercise	224
Qualifiers and Entity Types	231
Report Types	235

About This Guide

This guide describes how to begin using CONTROL-M to actively control jobs in the production environment.

Format of This Guide

Each chapter in this guide contains

- an introduction to the chapter and the material to be covered
- a review of the preparations necessary for performance of the exercises in the chapter
- an instruction set designed to explain the subject matter of the chapter
- a review and summary of the principal lessons in the chapter
- a list of recommended source material on the topics that were covered in the chapter

The exercises in this guide are designed to be progressive, with each exercise building on what was learned in a previous exercise. The chapters are organized the same way, with each building sequentially upon earlier material. A brief explanation of each chapter is shown below.

Chapter 1—Introduction to CONTROL-M

This chapter introduces you to the CONTROL-M Scheduling Definition facility and the Active Environment. It shows you the Scheduling Definition facility screens, the Active Environment screen, and some of the screens accessible from the Active Environment screen. In this chapter you will create two simple job scheduling definitions, and a dependency between them. You will then order these jobs, and observe and intervene in their processing.

Chapter 2–Scheduling Definition and Manual Intervention

This chapter provides a more detailed look at the parameters of the job scheduling definition, and introduces additional functionality in the Active Environment.

Chapter 3– Restarts under CONTROL-M/Restart

This chapter shows you how to define CONTROL-M/Restart parameters in the job scheduling definition, and how to intervene in the restart process when desired.

Chapter 4–Group Scheduling

This chapter introduces you to Group scheduling, and shows you how to define Group scheduling tables jobs.

Chapter 5–AutoEdit and JCL

This chapter introduces you to the AutoEdit facility, and has you use AutoEdit terms in both the JCL and in the job scheduling definition. In this chapter you will learn about and use system variables, local user defined variables, global variables, AutoEdit functions, and control statements.

Chapter 6–Navigating The Active Environment

This chapter takes you through features, facilities, and screens of the Active Environment that were not covered in the earlier chapters.

Chapter 7–Job Ordering and New Day Processing

This chapter instructs you in various ways of ordering jobs, and provides a detailed look at the use of the CTMJOB utility and User Daily jobs.

Chapter 8–Loose Ends

As the final chapter in this guide, this chapter describes various CONTROL-M and IOA facilities that were not covered in previous chapters. Included are the IOA Calendar facility, CMEM Event Management facility, and several useful utilities.

Appendix A–Sample JCL

This appendix lists all preparations that are necessary before you can perform the exercises in each chapter of this guide. It provides sample JCLs for those preparations, many of which involve the creation of JCL members.

Before You Begin

Before proceeding with the Getting Started Guide, you should read the introductory chapter of the *CONTROL-M for z/OS User Guide*.

You should perform the exercises in this guide in a test database.

It is entirely possible that other users will perform these exercises at the same time as you. To ensure that the someone else's exercises do not impact your exercises, you should use your unique user identifier as a prefix to all named components. In the text and screen examples in this guide, *ID* or *id* are used to indicate where your unique user identifier should be entered.

While it is not absolutely essential that you use your user ID as the unique prefix, it is absolutely essential that you consistently use the same brief, unique prefix of some sort if others will perform these exercises within the same time frame as you.

Conventions Used in This Guide

Notational conventions that may be used in this guide are explained below.

Standard Keyboard Keys

Keys that appear on the standard keyboard are identified in boldface, for example, **Enter**, **Shift**, **Ctrl+S** (a key combination), or **Ctrl S** (a key sequence).



WARNING

The commands, instructions, procedures, and syntax illustrated in this guide presume that the keyboards at your site are mapped in accordance with the EBCDIC character set. Certain special characters are referred to in this documentation, and you must ensure that your keyboard enables you to generate accurate EBCDIC hex codes. This is particularly true on keyboards that have been adapted to show local or national symbols. You should verify that

\$ is mapped to x'5B'
is mapped to x'7B'
@ is mapped to x'7C'

If you have any questions about whether your keyboard is properly mapped, contact your system administrator.

Preconfigured PFKeys

Many commands are preconfigured to specific keys or key combinations. This is particularly true with regard to numbered PF keys, or pairs of numbered PFKeys. For example, the END command is preconfigured to, and indicated as, **PF03/PF15**. To execute the END command, press either the **PF03** key or the **PF15** key.

Instructions to enter commands may include

- only the name of the command, such as, enter the END command
- only the PF keys, such as, press **PF03/PF15**
- or both, such as, press **PF03/PF15**, or enter the END command

Command Lines and Option Fields

Most screens contain a command line, which is primarily used to identify a single field where commands, or options, or both, are to be entered. These fields are usually designated **COMMAND**, but they are occasionally identified as **COMMAND/OPT** or **COMMAND/OPTION**.

Option field headings appear in many screens. These headings sometimes appear in the screen examples as **OPTION**, or **OPT**, or **O**.

Names of Commands, Fields, Files, Functions, Jobs, Libraries, Members, Missions, Options, Parameters, Reports, Subparameters, and Users

The names of commands, fields, functions, jobs, libraries, members, missions, options, parameters, reports, subparameters, users, and most files, are shown in standard **UPPERCASE** font.

User Entries

In situations where you are instructed to enter characters using the keyboard, the specific characters to be entered are shown in this **UPPERCASE BOLD** text, for example, type **EXITNAME**.

Syntax statements

In syntax, the following additional conventions apply:

- A vertical bar (|) separating items indicates that you must choose one item. In the following example, you would choose *a*, *b*, or *c*:

a | b | c

- An ellipsis (...) indicates that you can repeat the preceding item or items as many times as necessary.
- Square brackets ([]) around an item indicate that the item is optional. If square brackets ([]) are around a group of items, this indicates that the item is optional, and you may choose to implement any single item in the group. Square brackets can open ([) and close (]) on the same line of text, or may begin on one line of text and end, with the choices being stacked, one or more lines later.
- Braces ({}) around a group of items indicates that the item is mandatory, and you must choose to implement a single item in the group. Braces can open ({) and close (}) on the same line of text, or may begin on one line of text and end, with the choices being stacked, one or more lines later.

Screen Characters

All syntax, operating system terms, and literal examples are presented in this typeface. **This includes JCL calls, code examples, control statements, and system messages. Examples of this are:**

- calls, such as

```
CALL 'CBLTDLI'
```

- code examples, such as

```
FOR TABLE owner.name USE option, . . . ;
```

- control statements, such as

```
//PRDSYSIN DD * USERLOAD PRD(2) PRINT
```

- system messages, both stand-alone, such as You are not logged on to database database_name, and those embedded in text, such as the message You are not logged on to database database_name, are displayed on the screen.

Variables

Variables are identified with *italic* text. Examples of this are:

- In syntax or message text, such as
Specify database *database_name*
- In regular text, such as
replace database *database_name1* with database *database_name2* for the current session
- In a version number, such as
EXTENDED BUFFER MANAGER for IMS 4.1.xx

Special elements

This book includes special elements called *notes* and *warnings*:

NOTE



Notes provide additional information about the current subject.

WARNING



Warnings alert you to situations that can cause problems, such as loss of data, if you do not follow instructions carefully.

Help Information

Help information is readily available while using CONTROL-M. Help screens can be accessed by using the HELP command by typing **HELP** in the COMMAND line or by pressing **PF01/PF13** from any CONTROL-M screen.

Use standard scrolling conventions (**PF08/PF20** and **PF07/PF19**) to scroll forward and backward through the help information. To return to the original screen, use the END command (**PF03/PF15**).

Information New to This Version

Where substantive additions and modifications to the content of this guide occur, revision bars have been inserted in the margin.

Related Publications

CONTROL-M for z/OS Planning and Implementation Guide

Practical guide for determining implementation objectives and, for planning and performing the implementation of CONTROL-M

CONTROL-M for z/OS User Guide

Detailed explanation of the base CONTROL-M for z/OS product, an system that manages and automates the setup, scheduling and execution of jobs in the z/OS data center

INCONTROL for z/OS Administrator Guide

Information for system administrators about customizing and maintaining INCONTROL products

INCONTROL for z/OS Installation Guide

Step-by-step guide to installing INCONTROL™ products using the INCONTROL™ Installation and Customization Engine (ICE) application

INCONTROL for z/OS Messages Manual

Comprehensive listing and explanation of all INCONTROL and IOA messages and codes

INCONTROL for z/OS Security Guide

Step-by-step guide to implementing security in INCONTROL products

INCONTROL for z/OS Utilities Guide

Describes utilities designed to perform specific administrative tasks that are available to INCONTROL products

Introduction to CONTROL-M

This chapter includes the following topics:

Overview	24
Preparations	24
Main Components of CONTROL-M	24
Online Facility Primary Options	26
Introduction to the Scheduling Definition Facility	26
Introduction to the Active Environment Screen	36
Toggling Between Display of Commands and Options	36
Filtering the Active Environment Display	37
Confirming a Job Order	39
Checking the Log for the Job from the Active Environment Screen	41
Checking the Sysout for the Job	42
Holding and Deleting Active Environment Screen Job Orders	44
Return to the Scheduling Definition Facility	46
Saving Job Scheduling Definitions	46
Selecting a Table from the Table List Screen	48
Formats of the Job List Screen	50
Exiting the Online Facility	50
Review	51
Recommended Reading	52

Overview

CONTROL-M is an automated production control and scheduling system that manages and automates the setup, scheduling and execution of jobs in the z/OS data center.

The *CONTROL-M for z/OS Getting Started Guide* provides you with an introduction to the functionality of CONTROL-M, and teaches you how to use CONTROL-M, one step at a time, with each step building upon what was learned in the previous step. Of necessity, the explanations of topics in this book are not comprehensive. For comprehensive explanations of topics, you should refer to the relevant guide, such as the *CONTROL-M for z/OS User Guide*, *INCONTROL for z/OS Administrator Guide*, and so on.

The first chapter of the *CONTROL-M for z/OS User Guide* provides an introduction to CONTROL-M. You might find it useful to read that chapter before using this guide; however, you can still easily use this guide without having read that chapter in the User Guide.

This guide begins with a discussion of the main components of CONTROL-M:

- Scheduling Definition facility
- Tracking and Control facility: Active Environment screen

Before you are introduced to the main components of CONTROL-M, you should prepare the jobs that you will need in order to perform the exercises in this chapter.

Preparations

Create the JCL for two trivial jobs, identified throughout this guide as IDJOB1 and IDJOB2. Define the jobs to execute IEFBR14. For purposes of this guide, identify the JCL library as CTM.TEST.JCL.

Main Components of CONTROL-M

CONTROL-M automates the setup, scheduling, and execution of jobs in the z/OS environment. This is essentially a two step process:

1. You must tell CONTROL-M how you want job processing handled.

Job scheduling and processing instructions are provided through the Scheduling Definition facility, a main component of CONTROL-M.

You specify these instructions in job scheduling definitions that you define in the Job Scheduling Definition screen. Each job scheduling definition must be defined only once; it can then be used as often as needed. Job scheduling definitions are stored in scheduling tables, or members, in a scheduling library.

2. Using the instructions you provided in the job scheduling definitions, CONTROL-M automatically tracks and controls the setup, scheduling, and execution of the job. CONTROL-M provides you with constant feedback on job processing, and enables you to manually intervene in the processing of jobs, through a series of screens.

“Tracking and Control facility” is a general term that applies to the screens that provide feedback on job processing, and enable manual intervention. The main screen in this facility is the Active Environment screen, which provides status and other information for currently scheduled and recently completed jobs, and accepts your commands and instructions. The Active Environment screen also acts as a gateway to other screens in the Tracking and Control facility.

You will take a first look at these components by doing the following in your first set of exercises:

- You will use the Scheduling Definition facility to create job scheduling definitions for the two trivial jobs you just defined.
 - You will make submission of the first job dependent on a manual intervention performed in the Active Environment screen.
 - You will make submission of the second job dependent on successful completion of the first job.
- You will order the jobs.
- You will observe the results in the Active Environments screen. Along the way, you will become familiar with several commands, options and windows in the Active Environment screen. You will also become familiar with the CONTROL-M Log screen.

To do this, enter the IOA Online facility, take a look at the main components of CONTROL-M, and perform the exercises.

Online Facility Primary Options

The primary options of the Online facility are displayed when you enter the IOA Primary Option menu. Several of these options relate to CONTROL-M. Through most of this guide, you will concentrate on two options:

- Option 2, which provides access to the Scheduling Definition facility
- Option 3, which provides access to the Active Environment screen, the main screen of the Tracking and Control facility.

1 Request Option 2 from the IOA Primary Option menu, to access the Scheduling Definition facility.

The Scheduling Definition Facility entry panel is displayed.

Figure 1 CONTROL-M Scheduling Definition Facility - Entry Panel

```

----- CONTROL-M SCHEDULING DEFINITION FACILITY - ENTRY PANEL -----(2)
COMMAND ==>

SPECIFY LIBRARY, SCHEDULING TABLE, JOB

LIBRARY ==> CTM.TEST.SCHEDULE
TABLE ==> (Blank for table selection list)
JOB ==> (Blank for job selection list)

TYPE OF TABLE ==> ( J Job - default
                    G Group - for new tables only)

SHOW JOB DOCUMENTATION ==> N (Y/N)
AUTO-SAVE DOCUMENTATION ==> N (Y/N)

USE THE COMMAND SHPF TO SEE PFK ASSIGNMENT
23.00.04
    
```

Introduction to the Scheduling Definition Facility

You are about to create a CONTROL-M job scheduling definition for each of the jobs, IDJOB1 and IDJOB2, that you just defined. The JCL member name of the job should be used as the name of its corresponding job scheduling definition.

Job scheduling definitions are stored in scheduling tables (members) in a scheduling library. A default scheduling library name automatically appears in the entry panel. Ensure that you use a test library or your own work library. You should not use the JCL library.

For purposes of illustration in this guide, the scheduling library is identified as CTM.TEST.SCHEDULE. In this library, you will be creating a scheduling table called IDGS1. Use your unique identifier as the first characters of the scheduling table name.

You can now begin defining the job scheduling definition for IDJOB1.

- 2 Fill in the LIBRARY, TABLE, and JOB name fields in the entry panel. The request appears similar to [Figure 2](#).

Figure 2 CONTROL-M Scheduling Definition Facility - Entry Panel

```

----- CONTROL-M SCHEDULING DEFINITION FACILITY - ENTRY PANEL -----(2)
COMMAND ==>

SPECIFY LIBRARY, SCHEDULING TABLE, JOB

LIBRARY ==> CTM.TEST.SCHEDULE
TABLE   ==> IDGS1                (Blank for table selection list)
JOB     ==> IDJOB1              (Blank for job selection list)

TYPE OF TABLE      ==>          ( J Job - default
                                G Group - for new tables only)

SHOW JOB DOCUMENTATION ==> N      (Y/N)
AUTO-SAVE DOCUMENTATION ==> N    (Y/N)

USE THE COMMAND SHPF TO SEE PFK ASSIGNMENT                                     23.00.04

```

After entering the entry panel request, the Job Scheduling Definition screen is displayed. A full job scheduling definition, which contains more lines than your screen can display, is illustrated in [Figure 3 on page 28](#).

Note that the first line in the screen, which is displayed above the COMMAND line, contains the same information about the job that you provided in the entry panel.

Table 1 Job Scheduling Definition Screen Field Descriptions (Part 2 of 2)

Field	Description
DAYS	Days of the month to schedule the job.
CONFIRM	Yes/No indicator (Y/N) specifying whether manual confirmation is required before the job can be submitted.
OUT	Prerequisite conditions to be added and/or deleted when the job ends OK

3 Fill in the fields as indicated in the following table.

Table 2 Values for Job Scheduling Definition Screen Fields

Field	Value to Enter (in Bold), and Explanation
MEMNAME	IDJOB1 . This value, taken from the entry panel, already appears, because you specified the JCL member name in the JOB field of the entry panel.
MEMLIB	Name of the library containing JCL member <i>idJOB1</i> . For purposes of illustration, you should use CTM.TEST.JCL
GROUP	IDGRP1 . This name will later be used to filter the display of jobs in the Active Environment screen by Group name.
DESC	THIS JOB MUST BE RUN BEFORE IDJOB2 (to remind you what this job does).
DAYS	ALL (Schedule the job every day.)
CONFIRM	Y (Yes - you want manual confirmation for the job.)
OUT	IDJOB1-ENDED-OK ODAT + (After specifying IDJOB1-ENDED-OK, press Tab to get to the field in which to enter ODAT.)

The Job Scheduling Definition screen is displayed as shown in [Figure 4](#).


```

+-----+
MEMNAME IDJOB2      MEMLIB   CTM. TEST. JOBLIB
OWNER   M44         TASKTYPE JOB   PREVENT-NCT2 Y DFLT  N
APPL                                GROUP IDGRP1
DESC
OVERLIB                                STAT CAL
SCHENV                                SYSTEM ID      NJE  NODE
SET VAR
CTB STEP AT      NAME           TYPE
DOCMEM IDJOB2    DOCLIB
=====

```

8 In the IN field, define the same value you had defined as the OUT value in JOB1 (IDJOB1-ENDED-OK ODAT). You should press **Tab** before typing ODAT, as you did in IDJOB1. The + from the OUT value is not specified. After defining the IN value, delete the Y from the CONFIRM field.

```

=====
IN      IDJOB1-ENDED-OK      ODAT
CONTROL
RESOURCE
PIPE
FROM TIME      +      DAYS      UNTIL TIME      +      DAYS
DUE OUT TIME   +      DAYS      PRIORITY OO SAC CONFIRM
TIME ZONE:
=====

```

9 Change the plus sign (+) to a minus sign (-) in the OUT statement.

```

=====
OUT      IDJOB1-ENDED-OK      ODAT -
AUTO-ARCHIVE      SYSDB      MAXDAYS      MAXRUNS
=====

```

The IN and the OUT fields are used for specifying conditions. These conditions can be used to create job dependencies.

In this exercise, you used condition IDJOB1-ENDED-OK to ensure that IDJOB2 would not be submitted until IDJOB1 successfully completed:

- You made IDJOB1-ENDED-OK an IN condition for IDJOB2, which means that IDJOB2 cannot be run until this condition exists, that is, until IDJOB1-ENDED-OK has been created.
- You made IDJOB1-ENDED-OK an OUT condition of IDJOB1, specifying a plus sign in the appropriate place in the condition definition, which means that the condition will only be created after IDJOB1 completed successfully.

Therefore, IDJOB1 must complete successfully before IDJOB2 can be submitted.

Introduction to the Active Environment Screen

The Active Environment screen is your main interface to the Tracking and Control facility. It lists currently scheduled jobs. These jobs may or may not have already been submitted, and may or may not have already completed their executions.

Figure 9 CONTROL-M Active Environment Screen

```

Filter:          ----- CONTROL-M Active Environment ----- UP <D> (3)
COMMAND ==>>          SCROLL ==> CRSR
O Name      Owner      Odate  Jobname  JobID Typ ----- Status -----
CI CSPROD  M22          020201  CI CSPROD/04368  CST EXECUTING (RUN 1) GROUP=CI CS-PROD
CI CTEST   M22          020201  CI CTEST/04372  CST EXECUTING (RUN 2) GROUP=CI CS-TEST
BRI VPCC   I VP          020201  BRI VPCC /04369  JOB ENDED "OK"        GROUP=BR-I VP-CC
BRCC0001  I VP          020201  BRCC0001/04382  JOB ENDED "OK"        GROUP=BR-I VP-CC
BRCC0002  I VP          020201  BRCC0002/04383  JOB ENDED "OK"        GROUP=BR-I VP-CC
BRCC0003  I VP          020201  BRCC0003/04384  JOB ENDED "OK"        GROUP=BR-I VP-CC
BRCCI ND   I VP          020201  BRCCI ND /04385  JOB ENDED "OK"        GROUP=BR-I VP-CC
BRUPDT02  I VP          020201  BRUPDT02/04387  JOB ENDED "OK"        GROUP=BR-I VP-CC
BRREP001  I VP          020201  BRREP001/04388  JOB ENDED "OK"        GROUP=BR-I VP-CC
BRREP002  I VP          020201  BRREP002/04389  JOB ENDED "OK"        GROUP=BR-I VP-CC
BRI VPCCE  I VP          020201  /              JOB WAIT SCHEDULE     GROUP=BR-I VP-CC
CRCCEND   I VP          020201  /              JOB WAIT SCHEDULE     GROUP=BR-I VP-CC
INTRO001  M22          020201  /              JOB HELD WAIT SCHEDULE
                                         GROUP=INTER-PRODUCTI ON
INTRO002  M22          020201  /              JOB HELD WAIT SCHEDULE
                                         GROUP=INTER-PRODUCTI ON
INTRO003  M22          020201  /              JOB WAIT SCHEDULE
                                         GROUP=INTER-PRODUCTI ON
INTRO004  M22          020201  INTRO004/04371  JOB ENDED- NOT "OK" - ABENDED
Commands: OPt Display Show HIstory RBal REFresh Auto Jobstat SHPF Note Table
          OPt command toggles between Commands and Options display 15.15.48

```

This screen is most probably displaying a number of jobs from your site that are not relevant to these exercises. Soon, you will filter the display so that only the relevant jobs are displayed. But first, look at the list of available commands or options at the bottom of the screen.

Toggleing Between Display of Commands and Options

The bottom two lines of the Active Environment screen list either the commands or options available in the screen. Upon entry to the screen, the list of commands is displayed.

```

Commands: OPt Display Show HIstory RBal REFresh Auto Jobstat SHPF Note Table
          OPt command toggles between Commands and Options display 15.15.

```

The first command listed is OPT, and the bottom line actually explains that command OPT can be used to toggle between the command display and the option display.

- 15** Enter **OPT** in the **COMMAND** field to toggle to the option display. The list of available options is displayed at the bottom of the screen.

```
Opt: ? Why L Log H Hold Z Zoom R Rerun A Activate O Force OK V View Sysout
      N Net D Del F Free S Stat G Group U Undelete J JCL Edit C Confirm 15.46.0
```

- 16** Again, enter **OPT** in the **COMMAND** field to redisplay the list of available commands at the bottom of the screen.

```
Commands: OPt Display Show HIstory RBal REFresh Auto Jobstat SHPF Note Table
           OPt command toggles between Commands and Options display 15.15.48
```

One or the other of the two lists will be particularly relevant for different steps, and you will be instructed, as necessary, to display the appropriate screen. Now that the Command display appears, notice the **SHOW** command. You will be using the **SHOW** command in the next set of steps.

Filtering the Active Environment Display

As mentioned above, when you enter the active environment the screen is probably displaying a number of jobs from your site that are not relevant to these exercises. The following steps show how you can filter the display.

- 17** Enter **SHOW** in the **COMMAND** field. The Show Screen Filter window is displayed.

Figure 10 Show Screen Filter Window

```

----- Show Screen Filter -----(3. SHOW)-
Filter          Save  (Y/N) Desc:
Memname
Group
===== In Process Y ===== | Ended          Y | ===== State Y =====
-----
Wait Sched     Y Wait time Y | Ended "OK"    Y | Free          Y Forced OK Y
Wait Conf     Y Wait Cond  Y | Not "OK"     Y | Held          Y Grp Held  Y
Wait SUB      Y Wait quant Y | Rerun        Y | On Req       Y CMEM Forc Y
Submitted     Y Wait contrl Y | Disappeared  Y | Deleted      N Note      Y
Wait Exec     Y Grp Active Y | Abended      Y | Late         N Restarted Y
Executing     Y                | Unexpected CC Y | Pseudo       N
On Out Queue Y                | JCL Error    Y |
Task Type: Job Cyc Emr Stc Cst Est Ecj Ecs Wrn Grp
              Y  Y  Y  Y  Y  Y  Y  Y  Y  Y
Res Name
Resource Type: In Y Out Y Conds Y Resource Y Control Y
Owner
Odate:  From                To                Pri ori ty
Job
CPU Id          LPAR
Sch Li b
-----
OPt command toggles between Commands and Options display      11.07.49
    
```

The Show Screen Filter window enables you to filter the display based on certain fields and values. The following steps explain what you need to know in order to create the filter for use in these exercises. Filtering is explained in greater detail in “A Closer Look At Filters” on page 169.

- 18** Specify that you want to create and save a filter called IDGS by entering **IDGS** in the Filter field, and **Y** in the Save field. Specify a description for the filter by typing **GS-EXERCISES** in the Desc field.
- 19** Define the filtering criteria by specifying **IDGRP** in the Group field. The filter appears as shown below:

```

+----- Show Screen Filter -----(3. SHOW)+
| Filter IDGS      Save Y (Y/N) Desc: GS-EXERCISES |
| Memname                                     |
| Group IDGRP                                       |
    
```

The Group field works on a prefix basis. By assigning the IDGRP prefix value, the filter will only display jobs whose Group name begins with IDGRP. This means only the current jobs in group IDGRP1. In later chapters, it will also display jobs belonging to groups idGRP2, idGRP3, and so on.

- 20** Press **Enter** to exit the window. The Active Environment now displays only IDJOB1 and IDJOB2.

The log for IDJOB2 is displayed.

27 Press **PF03/PF15** again.

The Active Environment screen is redisplayed.

As an alternative to using the Log option in the Active Environment screen to request the log display for specific jobs, you can use Option 5 (LOG) of the IOA Primary Option menu to request display of the IOA Log screen. This screen displays the log for all INCONTROL for z/OS products. You can then filter the display in much the same way that you filtered the display of the Active Environment screen.

Checking the Sysout for the Job

If CONTROL-M/Restart is installed at your site, and the SYSDATA for a job is automatically archived, you can view the Sysout for that job.

To do so, you must first specify option **V** (View Sysout) for the job, in the Active Environment screen. This option displays the Job Order Execution History screen, which, as its name indicates, displays the execution history of the job. From the list of job executions, you can decide the job for which you want to view the Sysout.

To see the Sysout of a particular run of the job, specify option **S** (Select) next to the specific job execution in the Job Order Execution History screen. The sysout for the job is then displayed in the Sysout Viewing screen.

NOTE



If CONTROL-M/Restart is not operational at your site, or if SYSDATA is not archived, skip the remaining exercises in this section, and proceed to the topic “Holding and Deleting Active Environment Screen Job Orders” below.

Enter option **V** for job IDJOB1 to check the sysout of job IDJOB1.

The Job Order Execution History screen for IDJOB1 is displayed.

Figure 15 Job Order Execution History Screen

```

----- JOB ORDER EXECUTION HISTORY -----(3.V)
COMMAND ===>                                SCROLL===> CRSR
MEMNAME IDJOB1      OWNER ID                ORDERID 001S3  ODATE 020201
O JOBNAME  JOBID    DATE  START  ELAPSED  PAGES MAX RC  ----- STATUS  -----
S M21      32166   020201 18:05   0:00  00003      ENDED "OK"
===== >>>>>>>>> BOTTOM OF ACTIVE JOB ORDER HISTORY LIST <<<<<<<<<< =====

OPTION:  S SELECT                                02. 52. 14
    
```

The Job Order Execution History screen, and the Sysout Viewing screen shown below, are discussed in [Chapter 6, “Navigating The Active Environment.”](#)

28 Enter option **S** for the job run of IDJOB1 to request the desired Sysout.

The Sysout Viewing screen is displayed. You can scroll down and up, and right and left, through the Sysout.


```
CTM659I DELETE OF TASK IDJOB2 GROUP "IDGRP1 " ODATE 020201 PERFORMED (3)
COMMAND ===>                                     SCROLL ==> CRSR
O Name      Owner      Odate   Jobname   JobID    Typ ----- Status -----
===== >>>>>>>>>>>>>> Bottom of Jobs Li st <<<<<<<<<<<<<< =====
```

Do not confuse deleting a job order from the Active Environment screen with deleting a job scheduling definition from the Job List screen. Deleting the job order impacts only that order. The job scheduling definition itself remains unaffected and can be ordered again. If you delete a job scheduling definition from the Job List screen, the job scheduling definition itself is gone, and the job can no longer be ordered from the scheduling table.

Return to the Scheduling Definition Facility

In this chapter, you will do a little more work in the Scheduling Definition facility.

Saving Job Scheduling Definitions

So far, you have created and exited two job scheduling definitions—for IDJOB1 and IDJOB2. But they are currently only in memory; you have not saved them.

You can now save the job scheduling definitions. To do that you must first return to the Scheduling Definition facility.

35 Enter =2 in the COMMAND field.

The Job List screen is displayed.

36 Exit the Job List screen.

The Exit option window is displayed. This window prompts whether you want to save the job scheduling definitions you created.

Figure 18 Job List Screen Exit Option Window

JOB LI ST	LIB: CTM. TEST. SCHEDULE	TABLE: I DGS1
COMMAN	=====	
OPT N	PLEASE SELECT EXIT OPTION	====> CRSR
i		-----
i	SAVE CREATE	
=====		<<< =====
	LIBRARY CTM. TEST. SCHEDULE	
	TABLE I DGS1	

OPTIONS S SEL D DEL I INS O ORDER F FORCE J JCL C COPY P PLN T JOBSTAT 16. 22. 13		

The **LIBRARY** and **TABLE** fields indicate the library and table in which the job scheduling definitions were defined. The specified values can be modified, for example, to save the job scheduling definitions in a different table.

The window also contains **SAVE** and **CREATE** fields. The Exit Option window enables you to perform many actions.

- To create a new table in a library and save the job scheduling definitions in that table, specify **Y** (Yes) in the **CREATE** field of the window, leaving the **SAVE** field blank, and exit the window.

Because you specified a new table name in the entry panel—one that had not previously existed—you are now going to permanently create that table as well. Therefore, the values you entered in this chapter are the values that you will use.

If you changed the library and table fields of the window to reflect a currently non-existing table, you would also use these values.

- To save the job scheduling definitions in an existing table in a library, specify **Y** (Yes) in the **SAVE** field of the window, leaving the **CREATE** field blank, and exit the window. This procedure should be used even if you change the library and or table fields to a different, but existing, table.
- To cancel the changes you made, specify **N** (No) in the both the **SAVE** and **CREATE** fields of the window, and exit the window.

- To close the Exit Option window and remain in the Job List screen, with the changes remaining in memory, press **PF04/PF16** to execute the RESET command.

37 Specify Y in the CREATE field and leave the SAVE field blank.

The new scheduling table is created and the job scheduling definitions are saved in the table. The Entry Panel is displayed.

NOTE



Normally, when you exit the Job List screen, the Table List screen is displayed. However, if the Table List screen was bypassed when you entered the Scheduling Definition facility, that is, if you specified a TABLE value in the entry panel, which you did, the Table List screen is not displayed upon exiting the Job List screen; instead, the Entry Panel is displayed.

Selecting a Table from the Table List Screen

You can now enter the Table List screen so that it displays your newly created table.

38 Ensure that correct library name is entered in the LIBRARY field of the entry panel and that the TABLE field is blank.

The Table List screen is displayed. Unless you are using a library dedicated to this set of exercises, other table names appear in the Table list screen.

Figure 19 CONTROL-M Table List Screen

LIST OF TABLES IN CTM. TEST. SCHEDULE										----- (2)	
COMMAND ==>										SCROLL==> CRSR	
OPT	NAME	VV	MM	CREATED	CHANGED	SIZE	INIT	MOD	ID		
	ADABAS	01	00	98/09/09	00/05/09 00:50	70	70	0	001		
	APPLNTN	01	00	98/09/09	00/05/09 00:50	180	180	0	001		
	APPLPRDI	01	00	98/09/09	00/05/09 00:50	41	41	0	001		
	ARCNI GHT	01	00	98/09/09	00/05/09 00:50	5	5	0	S07		
	ASMBTR1	01	00	98/09/09	00/05/09 00:50	9	9	0	S07		
	ASMBTR2	01	00	98/09/09	00/05/09 00:50	14	14	0	S07		
	BACKUP	01	00	98/09/09	00/05/09 00:50	5	5	0	S07		
	CI CSJOBS	01	00	98/09/09	00/05/09 00:50	70	70	0	001		
	CI CSPROD	01	00	98/09/09	00/05/09 00:50	180	180	0	001		
	CI CSTEST	01	00	98/09/09	00/05/09 00:50	41	41	0	001		
	CI CSUPT	01	00	98/09/09	00/05/09 00:50	5	5	0	S07		
	CLI ENTS	01	00	98/09/09	00/05/09 00:50	9	9	0	S07		
	DB2EXE	01	00	98/09/09	00/05/09 00:50	14	14	0	S07		
	DLOAD	01	00	98/09/09	00/05/09 00:50	5	5	0	S07		
	MAI NDAY	01	00	98/09/09	00/05/09 00:50	180	180	0	001		
	MAI NT	01	00	98/09/09	00/05/09 00:50	41	41	0	001		
	MAI NTPL	01	00	98/09/09	00/05/09 00:50	5	5	0	S07		
	ONSPPOOL	01	00	98/09/09	00/05/09 00:50	9	9	0	S07		
	ONSPPOOLT	01	00	98/09/09	00/05/09 00:50	14	14	0	S07		
	OPERCLN	01	00	98/09/09	00/05/09 00:50	5	5	0	S07		
OPTIONS S SELECT O ORDER F FORCE G GRAPHIC FLOW B BROWSE D DELETE 15.38.37											

Formats of the Job List Screen

You had previously used the DESC command to ensure the desired format of the screen. This command confirmed that you wanted the display type that shows the description of the job, as taken from the DESC field in the job scheduling definition, to be displayed next to the job name.

Table 3 describes the commands used to display other information.

Table 3 Commands to Display Job List Screen Formats

Command	Description
DATA	Command DATA displays the Application name and Group name of the job next to the job name. The Application name and Group name are taken from the corresponding fields in the job scheduling definition.
STAT	Command STAT displays, next to the job name, the following ISPF-like statistical information about the job: Version and modification numbers, creation date, last modification date, and user ID.

Alternate between specifying the **DATA**, **STAT**, and **DESC** commands in the COMMAND field of the Job List screen, and note the differences in the display following the entry of each command. Ensure that the last command you enter is **DESC**, because that is the format you are using in the exercises in this guide.

Note that whichever format you use, the actual jobs listed in the screen, that is, jobs IDJOB1 and IDJOB2, remain the same. The format of the Job List screen does not affect which jobs are listed. It only affects the accompanying data that is displayed for those jobs.

Exiting the Online Facility

You have now finished the most of the exercises in this chapter of the guide. All that remains is to learn to exit the IOA Online facility.

Leave the online facility by exiting one screen at a time, until you have exited the entire online facility.

- 41** Press **PF03/PF15** or enter the END command.

The Table List screen is displayed.

- 42** Press **PF03/PF15** or enter the END command.

The Entry panel is displayed.

- 43** Press **PF03/PF15** or enter the END command.

The IOA Primary Option Menu is displayed.

- 44** Enter **X** in the COMMAND field of the IOA Primary Option menu.

The IOA Online facility is exited.

Review

In this chapter you

- accessed the following screens of the scheduling definition facility
 - Entry Panel
 - Table List screen
 - Job List screen
 - Job Scheduling Definition screen
- created a job scheduling definition directly from the Entry panel, and then created a second job scheduling definition by using the I (Insert) option in the Job List screen
- learned about and utilized the following job scheduling definition fields while creating your job scheduling definitions
 - MEMNAME
 - MEMLIB
 - GROUP
 - DESC
 - DAYS
 - IN
 - CONFIRM
 - OUT
- defined one of the jobs to require manual confirmation before submission
- used IN and OUT conditions to create a dependency between the two jobs
- ordered the two jobs from the Job List screen
- viewed messages in the Job Order Messages screen

- moved between facilities using an =x command instead of returning to the IOA Primary Option menu
- displayed the Active Environment screen, and toggled between display of the list of commands and the list of options at the bottom of the screen
- defined a filter and filtered the Active Environment display
- learned about statuses in the Active Environment screen
- used the C (Confirm) option in the Active Environment screen to manually confirm a job
- observed changes to job statuses in the Active Environment screen
- used the L (Log) option in the Active Environment screen to display the message log for the job orders in the CONTROL-M Log screen
- used the V option to display the Job Order Execution History screen and from there displayed the Sysout of a job execution in the Sysout Viewing screen
- used the H (Hold) option so that you could delete job orders from the Active Environment screen
- used the D (Delete) option in the Active Environment screen to delete the job orders when they were no longer needed
- used the Exit Option window of the Job List screen to create a new scheduling table and save your two job scheduling definitions in the table
- learned the difference between the S (Select) and B (Browse) options of the Table List screen
- changed the format of the Job List screen
- exited the Online facility by working your way up the screen hierarchy as you exited individual screens

Recommended Reading

Before continuing with the next chapter, you should read the following in the *CONTROL-M for z/OS User Guide*

In Chapter 1

- Introduction to CONTROL-M components
- Prerequisite condition concept
- Date concepts

In Chapter 2

- Introduction to the Scheduling Definition Facility
- Description of the following Scheduling Definition Facility screens including their formats, options and commands
 - Entry Panel
 - Table List screen
 - Job List screen
- Description of the layout of the Job Scheduling Definition screen and the fields in the screen
- Topics “Exiting the Scheduling Definition Facility” and “Ordering (Scheduling) Jobs” in the Scheduling Definition facility description
- Introduction to the Active Environment screen and the description of its fields
- Descriptions of the CONTROL-M and IOA Log screens

In Chapter 3

Detailed descriptions of the parameters you have already used: CONFIRM, DAYS, DESC, GROUP, IN, MEMLIB, MEMNAME, OUT



NOTE

The description of the parameter DAYS is quite detailed. Read it for a general understanding of parameter usage and possible parameter formats, without trying to memorize all the detail. For now, you can skip the information related to subparameters DCAL and AND/OR.

Scheduling Definition and Manual Intervention

This chapter includes the following topics:

Overview	56
Preparations	56
Sections of the Job Scheduling Definition	57
General Parameters Section	59
Basis Scheduling Parameters Section	60
Runtime Scheduling Parameters Section	64
Post-Processing Parameters Section	67
Conditions, Date Qualifiers, and Job Dependencies	70
Displaying the Scheduling Plan for the Job	71
Defining JOB4	74
Job Ordering and Job Forcing	76
Why Screen	78
Why the Job Did Not Run	79
Adding a Missing Condition in the Why Screen	80
IOA Conditions/Resources Screen	81
Filtering the IOA Conditions/Resources Screen	81
Adding a Condition or Resource	82
A Second Look at the Why Screen	83
Zoom Screen	84
Freeing a Held Job	86
Review	88
Recommended Reading	89

Overview

In the previous chapter, you were introduced to the Scheduling Definition Facility and the Active Environment screen. In this chapter, you will

- take a much closer look at the job scheduling definition screen, the sections into which it is divided, and the many of the fields of those sections, including a focus on
 - Basic Scheduling parameters
 - Runtime Scheduling parameters
 - Post-Processing parameters
- view the scheduling plan of a job in a graphic calendar display
- force a job whose basic scheduling criteria are not satisfied
- use the Why screen, IOA Conditions/Resources screen, and the Zoom screen to intervene when a job is not submitted

Preparations

In this chapter, you are going to create two new jobs and their job scheduling definitions. In the last chapter, you used the names IDJOB1 and IDJOB2. In this chapter, you will use the names IDJOB3 and IDJOB4, respectively. (You will use the same library and table that you used in the last chapter.)

For the JCL of IDJOB4, use a copy of the same trivial JCL that you used for IDJOB1 & IDJOB2—that is, a job that does basically nothing, but ends successfully.

For IDJOB3, you need a JCL that will result in an error condition. For this purpose, you will use a test utility in which you indicate the desired error return code.

You can now create the JCL for those jobs.

- Create trivial job IDJOB3 to run utility TESTUTIL and define it to return a condition code of C0008.
- Create trivial job IDJOB4 and define it to run IEFBR14.

Sections of the Job Scheduling Definition

- 1 Enter the IOA Online Facility and request the Scheduling Definition facility.

The Scheduling Definition entry panel is displayed.

- 2 Enter your library and table names in the LIBRARY and TABLE fields, and IDJOB3 in the JOB field. The request appears as shown below:

Figure 21 CONTROL-M Scheduling Definition Facility - Entry Panel

```

----- CONTROL-M SCHEDULING DEFINITION FACILITY - ENTRY PANEL -----(2)
COMMAND ===>

SPECIFY LIBRARY, SCHEDULING TABLE, JOB

LIBRARY ===> CTM. TEST. SCHEDULE
TABLE   ===> IDGS1           (Blank for table selection list)
JOB     ===> IDJOB3         (Blank for job selection list)

TYPE OF TABLE      ===>      ( J Job - default
                               G Group - for new tables only)

SHOW JOB DOCUMENTATION  ===> N      (Y/N)
AUTO-SAVE DOCUMENTATION ===> N      (Y/N)

USE THE COMMAND SHPF TO SEE PFK ASSIGNMENT                                23.00.04

```

The Job Scheduling Definition screen for IDJOB3 is displayed.

Figure 22 Job Scheduling Definition Screen

```

JOB: IDJOB3   LIB CTM. TEST. SCHEDULE           TABLE: IDGS1
COMMAND ==>>                                SCROLL==>> CRSR
-----+-----
MEMNAME IDJOB3      MEMLIB
OWNER   ID          TASKTYPE JOB    PREVENT-NCT2  DFLT  N
APPL
DESC
OVERLIB
SCHENV          SYSTEM ID              STAT CAL
SET VAR
CTB STEP AT      NAME          TYPE
DOCMEM IDJOB3    DOCLIB
=====
DAYS
WDAYS
MONTHS  1- Y 2- Y 3- Y 4- Y 5- Y 6- Y 7- Y 8- Y 9- Y 10- Y 11- Y 12- Y
DATES
CONFCAL          RETRO N MAXWAIT 00  D-CAT
MINIMUM          PDS
DEFINITION ACTIVE FROM          UNTIL
=====
RESOURCE
COMMANDS: EDIT, DOC, PLAN, JOBSTAT
11. 17. 00
    
```

In Chapter 1, “Introduction to CONTROL-M.”, you learned that the job scheduling definition is partitioned into sections, each section separated from the preceding section by a line of equal signs.

To familiarize yourself with several more fields in these sections, you will fill in the job scheduling definition for IDJOB3. To do this, you must first outline the type of processing you expect for IDJOB3, which requires that you assume the following:

- IDJOB3 is an application that tabulates certain fiscal results once a month.
- IDJOB3 should always be scheduled on the last Friday of the month.
- IDJOB3 should be submitted only if the previous execution of the job (during the previous month) ended OK.
- IDJOB3 should be submitted only if the following resources are available:
 - 1 tape drive
 - 5 units of a CPU resource called IDCPOWER:
- IDJOB3 should be submitted from sometime between the hours of 11:00 p.m. on Friday night and 4:00 a.m. on Saturday Morning.
- IDJOB3 should be considered a critical path job with a priority of 5.

- IDJOB3 should be automatically submitted, no requirement for manual confirmation.
- If IDJOB3 ends OK, the appropriate reports are generated by the application.
- If the budget for IDJOB3, as defined in the application, was exceeded, the job returns a condition code of C0008 or above, depending on the amount by which the budget was exceeded, and the job should end NOTOK.
- If IDJOB3 ends OK, a prerequisite condition should be created indicating that it ended OK and the corresponding condition from the previous month's run should be deleted.
- If IDJOB3 ends NOTOK, that is, if it has a condition code of C00088 or higher, the following should occur:
 - A condition should be created stating that the job ended NOTOK.
 - An appropriate message should be sent, or shouted, to user OPER1.
 - An auxiliary job, called IDJOB4, should be run to perform additional processing.

You can now fill in the job scheduling definition of IDJOB3. Note that fields are described only briefly in the following sections, and that not all fields are discussed. Detailed descriptions for all fields of the Job Scheduling Definition screen are discussed in the *CONTROL-M for z/OS User Guide*.

General Parameters Section

The fields in the first section of the Job Scheduling Definition screen correspond to the Job Scheduling Definition General parameters.

MEMNAME	IDJOB3	MEMLI B	PREVENT-NCT2	DFLT	N
OWNER	ID	TASKTYPE	JOB	GROUP	
APPL					
DESC					
OVERLI B					STAT CAL
SCHENV			SYSTEM	ID	NJE NODE
SET VAR					
CTB STEP	AT	NAME		TYPE	
DOCMEM	IDJOB1	DOCLI B			

The General parameters, some of which you have seen earlier, provide basic information about the job, and are shown in the following table.

Table 4 Job Scheduling Definition Screen General Parameters

Parameters	Type of Information
MEMNAME, MEMLIB	Location of the JCL
OWNER	Owner of the job
TASKTYPE	Type of job
DESC	Job description
APPL, GROUP	Application and group to which the job belongs
DOCMEM, DOCLIB	Location of documentation about the job

- 3 Specify the name of your test JCL library in the MEMLIB field
- 4 Specify **IDGRP2** in the GROUP field.
- 5 Specify **FISCAL RESULTS ON LAST FRIDAY OF MONTH** in the DESC field. The section looks as follows:

```

+-----+
MEMNAME  IDJOB3      MEMLIB  CTM. TEST. JCL
OWNER    ID          TASKTYPE JOB  PREVENT-NCT2  DFLT  N
APPL                                           GROUP  IDGRP2
DESC     FISCAL RESULTS ON LAST FRIDAY OF MONTH
OVERLIB
SCHENV                               SYSTEM ID          STAT CAL
SET VAR
CTB STEP AT          NAME          TYPE
DOCMEM  IDJOB3      DOCLIB
=====
    
```

Basis Scheduling Parameters Section

The fields in the second section of the Job Scheduling Definition screen correspond to the Job Scheduling Definition Basic parameters.

```

=====
DAYS                                           DCAL
                                           AND/OR
WDAYS                                           WCAL
MONTHS  1- Y 2- Y 3- Y 4- Y 5- Y 6- Y 7- Y 8- Y 9- Y 10- Y 11- Y 12- Y
DATES
CONFICAL                                RETRO N MAXWAIT 00  D-CAT
MINIMUM          PDS
DEFINITION ACTIVE FROM          UNTIL
=====
    
```

These basic scheduling parameters determine on what day the job is scheduled, specifically, on what days the job becomes a candidate for execution. Scheduling days can be expressed in many ways, according to need. Several of the basic scheduling parameters are explained in the following table:

Table 5 Basic Job Scheduling Definition Parameters

Parameter	Explanation
DAYS	<p>Days within the specified months on which to schedule the job. For more information, see MONTHS in this table.</p> <p>Values can be expressed in various formats, the simplest of which is “straight values.” For example, values 01 and 15 would mean the first and fifteenth day of the month, respectively.</p> <p>Value ALL means all days of the month.</p>
WDAYS	<p>Days of the week within the specified months on which to schedule the job. For more information, see MONTHS in this table.</p> <p>Values can be expressed in various formats.</p> <p>Scheduling is relative to the day of the week that is defined as the start of the week at your site. For example, if Monday is defined as the start, or first day of the week at your site, a value of 1 means that the job is scheduled on Mondays, and 4 means that the job is scheduled on Thursdays.</p>
AND/OR	<p>Conjunctural parameter that determines the relationship between the DAYS and WDAY parameters. An A (And) value requires that both sets of criteria be satisfied. An O (Or) value means that the criteria are satisfied if the DAYS criteria and/or the WDAY criteria are satisfied.</p>
MONTHS	<p>Months in which to schedule the job.</p> <p>All twelve months are listed as 01 through 12. A Y (Yes) or N (No) value can be specified for each month.</p> <p>For months in which a job should be scheduled, that is, months with a Y value, the job is scheduled on those days determined by the DAYS and/or WDAY values.</p>
DATES	<p>Specific dates on which to schedule the job. Format is mmdd or ddmm, depending on the site standard. For example, a DATES value of 0505 schedules the job on the 5th of May.</p>

Several examples of how these parameters might be used are shown below.

- To schedule a job on the 1st and 15th of every month

```

=====
DAYS      01, 15                                DCAL
                                                AND/OR
WDAYS
WCAL
MONTHS    1- Y 2- Y 3- Y 4- Y 5- Y 6- Y 7- Y 8- Y 9- Y 10- Y 11- Y 12- Y
DATES
CONFCAL          RETRO N MAXWAIT 00  D-CAT
MINIMUM          PDS
DEFINITION ACTIVE FROM          UNTIL
=====

```

- To schedule a quarterly job on 31 March, 30 June, 30 September and 31 December (assuming a date format of mmdd)

```

=====
DAYS
                                                DCAL
                                                AND/OR
WDAYS
WCAL
MONTHS    1- Y 2- Y 3- Y 4- Y 5- Y 6- Y 7- Y 8- Y 9- Y 10- Y 11- Y 12- Y
DATES      0331 0630 0930 1231
CONFCAL          RETRO N MAXWAIT 00  D-CAT
MINIMUM          PDS
DEFINITION ACTIVE FROM          UNTIL
=====

```

- To schedule a job on every Wednesday in June (assuming start of week is Monday)

```

=====
DAYS
                                                DCAL
                                                AND/OR
WDAYS      3                                WCAL
MONTHS    1- N 2- N 3- N 4- N 5- N 6- Y 7- N 8- N 9- N 10- N 11- N 12- N
DATES
CONFCAL          RETRO N MAXWAIT 00  D-CAT
MINIMUM          PDS
DEFINITION ACTIVE FROM          UNTIL
=====

```

In your job scheduling definition, you want to schedule the job on the last Friday of the month. For this you need to know that in the DAYS parameter, a value of L1 means the last day of the month, L2 is the second to the last day of the month, and so on.

- 6 In the DAYS field, specify the following values: **L1,L2,L3,L4,L5,L6,L7**. Be sure to separate values by a comma, and leave no embedded spaces.
- 7 In the AND/OR field, specify **A** (And).
- 8 In the WDAY5 field, specify **5**.



NOTE

This example assumes your site-defined start of week is Monday. If the start of week is defined as Sunday, use a WDAYs value of **6** instead of **5**. If you do not know the site-defined start of week, it does not really matter which value you use in this exercise. But to use this parameter successfully in your real job scheduling definitions, you need to know, and should find out from your INCONTROL administrator, your site-defined start of week.

- 9 Make no changes to the MONTHS values, because they are correctly set to **Y** (Yes) for all months.

This section of your screen appears as shown below:

DAYS	L1, L2, L3, L4, L5, L6, L7	DCAL	
		AND/OR	A
WDAYS	5	WCAL	
MONTHS	1- Y 2- Y 3- Y 4- Y 5- Y 6- Y 7- Y 8- Y 9- Y 10- Y 11- Y 12- Y		
DATES			
CONF CAL		RETRO N	MAXWAIT 00 D-CAT
MI NI MUM	PDS		
DEFI NI TI ON	ACTI VE FROM		UNTI L

Note the following points:

- The values of the DAYS parameter limits the scheduling to the last week of the month.
- The value of the WDAYs parameter limits the scheduling to Fridays.
- The A (And) value in the AND/OR field requires that both sets of criteria be satisfied. This limits the scheduling to the last Friday in the last week of the month.
- The Y (Yes) value for all months ensures that the scheduling on the last Friday of the month will occur every month.

Runtime Scheduling Parameters Section

The fields in the third section of the Job Scheduling Definition screen correspond to the Job Scheduling Definition Runtime Scheduling parameters.

```

=====
I N
CONTROL
RESOURCE
PIPE
FROM TIME      +   DAYS   UNTIL TIME      +   DAYS
DUE OUT TIME   +   DAYS   PRIORITY      SAC   CONFIRM
TIME ZONE:
=====
    
```

NOTE



The PIPE parameter is only displayed if MAINVIEW Batch Optimizer (MVBO) is installed at your site.

The Runtime Scheduling parameters define runtime criteria that must be satisfied before the job can be submitted. You have already seen the IN parameter and CONFIRM parameters. [Table 6](#) explains several of the Runtime Scheduling parameters.

Table 6 Runtime Scheduling Parameters (Part 1 of 2)

Parameter	Explanation
CONTROL	Control Resource. Each Control resource specification consists of two values: Resource name and the type of control required. Valid values for the type of control are: <ul style="list-style-type: none"> ■ E – Exclusive ■ S – Shared For example, this parameter can be used to specify that exclusive control of a particular file is required. CONTROL-M does not submit the job until it has exclusive control of that file.
RESOURCE	Quantitative Resource. Each Quantitative resource specification consists of two values: Resource name and the quantity required. For example, this parameter can be used to specify that two tape drives are required by the job. CONTROL-M does not submit that job until two tape drives are available and reserved for that job.

Table 6 Runtime Scheduling Parameters (Part 2 of 2)

Parameter	Explanation
TIME FROM UNTIL	<p>TIME FROM indicates the earliest time that CONTROL-M can submit the job. If specified, the job is not submitted before this time of the scheduling day.</p> <p>TIME UNTIL indicates the latest time by which a job can be submitted. If CONTROL-M is unable to submit the job before this time (for example, if a required resource is not available before the UNTIL time), the job is not submitted.</p> <p>Either value or both, or neither, can be specified. No value means that a time factor is irrelevant for job submission.</p> <p>Note: It is possible to specify the FROM/TO offset in days in addition to the FROM/TO time. For details, see the <i>CONTROL-M for z/OS User Guide</i>.</p>
PRIORITY	<p>Job priority. If two or more jobs awaiting submission are competing for resources, this value enables CONTROL-M to determine which job to submit, based on priority.</p> <p>Alphabetic or numeric values can be specified, where $A < Z < 1 < 9$.</p> <p>Critical path priority can be indicated by specifying an asterisk (*) before the priority value (for example, *5). A critical path job always takes precedence over a non-critical path job.</p>

You had previously determined the following submission criteria for the job:

- The job can be submitted only if the previous month's execution of the job ended OK.
- The job requires the following resources:
 - 1 tape drive
 - 5 units of CPU
- The job must be submitted between the hours of 11:00 p.m. (Friday) and 4:00 a.m. (Saturday).
- The job is a critical path job with a priority of 5.
- The job must be submitted automatically without manual confirmation.

You should now define these criteria.

10 In the IN fields, specify **IDJOB3-ENDED-OK** as the condition name.

11 Press **Tab** after specifying the condition name, to place the cursor in the DATE field.

12 Specify **PREV** as the value for the DATE field.

When you define post-processing parameters, you will see that when the job ends OK this same IN condition is added, but with a date of ODAT.

13 In the RESOURCE field, specify **IDTAPE-DRIVE** as the resource name.

14 Press **Tab** after specifying the resource name to place the cursor in the QUANTITY field.

15 Specify **1** as the value for the QUANTITY field.

Two resources can be specified on a single line. Once you specify the quantity of the first resource, the cursor automatically tabs over to the RESOURCE-NAME field for the second resource.

16 Specify **IDCPOWER** as the second resource name, and **5** as the quantity.

Once you have filled in two resources on the resource line, an additional resource line is opened to enable you to specify more resources.

17 In the TIME FROM field, specify **2300**. In the UNTIL field specify **0400**.

18 In the PRIORITY field, specify ***5**.

19 In the CONFIRM field, which you defined in the previous exercise, delete the **Y** value.

This section of the job scheduling definition appears as shown below:

```

=====
IN          I DJOB3-ENDED-OK      PREV
CONTROL
RESOURCE I DTAPE-DRIVE           0001      IDCPOWER           0005
PIPE
FROM TIME 2300 + DAYS UNTIL TIME 0400 + DAYS
DUE OUT TIME + DAYS PRIORITY *5 SAC CONFIRM
TIME ZONE:
=====

```


Table 7 Post-Processing Parameters ON and DO Statements (Part 2 of 2)

Parameter	Explanation
DO	<p>DO statements specify actions to be performed if the ON criteria are satisfied. Many different DO statements are available. For example:</p> <p>DO COND adds or deletes a prerequisite condition.</p> <p>DO FORCEJOB forces a job to be scheduled.</p> <p>DO SHOUT sends, or shouts, a message to a specified destination.</p> <p>DO RERUN reruns the job.</p> <p>When you type the name or abbreviation of the DO statement, such as COND, or SHOUT, in the DO field, the appropriate subparameter fields for the statement are displayed.</p> <p>If by mistake you request the wrong DO statement, you can delete it by selecting the DO statement name, such as COND, and pressing Enter.</p>

Note that this section contains a SHOUT WHEN statement. This statement, which is not examined in this chapter, is different than the DO SHOUT statement.

You had previously decided to request the following post-processing activities:

- If the job ends OK, which is a condition code of less than C0008
 - the prerequisite condition of the successful run from the previous month, should be deleted
 - a new prerequisite condition indicating that the job ended OK should be created.
- If the job ends NOTOK, that is, it has a condition code of C0008 or higher
 - a condition must be created to indicate that the job ended NOTOK.
 - an appropriate message must be shouted to your user ID.
 - auxiliary job IDJOB4 must be run to perform additional processing.

You can now define the necessary values.

20 Fill in the OUT fields with the following value:
 IDJOB3-ENDED-OK ODAT +

After filling in the second condition, a new, blank OUT condition line is opened to enable you to specify another condition.

21 In the ON block, specify **ANYSTEP** in the PGMST (program step) field, and specify the value **>C0007** in the CODES field. A value equal to or greater than C0008 is synonymous with a value greater than C0007.

22 In the DO field, type **COND** and press **Enter**.

The DO COND statement is opened, as is a blank DO statement for specification of another DO statement.

23 Fill in the DO COND fields with the following values:

I DJOB3-END-NOTOK ODAT +

24 In the blank DO field, type **SHOUT** and press **Enter**.

The DO SHOUT statement is opened, as is a blank DO statement for specification of another DO statement, as shown below.

DO SHOUT	TO	URGENCY
=		
DO		

25 In the DO SHOUT statement, specify your user ID in the TO subparameter field; replace the **R** (Regular) with **U** (Urgent) in the URGENCY field; and specify the message **FRIDAY FISCAL RESULTS RUN FAILED** following the equal sign (=).

Note that the only urgency value, other than **R** and **U**, is **V**, which signifies a Very Urgent message.

26 In the blank DO field, type **FORCEJOB** and press **Enter**.

The DO FORCEJOB statement is opened, as is a blank DO statement for specification of another DO statement, as shown below.

DO FORCEJOB	TABLE	JOB	DATE
	LI BRARY		
DO			

27 In the DO FORCEJOB statement, specify the current scheduling table name (**IDGS1**) in the TABLE field; specify **IDJOB4** in the JOB field; specify **ODAT** in the DATE field; and specify the current scheduling library name in the LIBRARY field.

This section of the job scheduling definition appears as shown below:

```

=====
OUT      IDJOB3-ENDED-OK      ODAT +

AUTO-ARCHIVE      SYSDB      MAXDAYS      MAXRUNS
RETENTION: # OF DAYS TO KEEP      # OF GENERATIONS TO KEEP
SYSOUT OP (C, D, F, N, R)
MAXRERUN      RERUNMEM      INTERVAL      FROM
STEP RANGE      FR (PGM. PROC)      TO      FROM
ON PGMST ANYSTEP      PROCST      CODES >C0007      A/O
DO COND      IDJOB3-END-NOTOK      ODAT +
DO SHOUT      TO ID      URGENCY U
= FRI DAY FISCAL RESULTS RUN FAILED
DO FORCEJOB TABLE IDGS1      JOB IDJOB4      DATE ODAT
LI BRARY CTM. TEST. SCHEDULE

DO
ON PGMST      PROCST      CODES      A/O
DO
    
```

You have now finished defining IDJOB3.

28 Exit the job scheduling definition of IDJOB3.

The Job List screen is displayed. This is where you can take a closer look at the use of IN, OUT, and DO COND conditions, and their date qualifiers, to create job dependencies.

Conditions, Date Qualifiers, and Job Dependencies

Conditions are generally used to create job dependencies. In the previous chapter you saw that when IDJOB1 ended OK, it set an OUT condition (IDJOB1-ENDED-OK) that satisfied the prerequisite IN condition to IDJOB2. This is necessary because IDJOB2 could not be submitted until IDJOB1 ended OK.

Earlier in this chapter you defined a DO COND condition. Both OUT and DO COND are post-processing statements that add or delete conditions. But the OUT condition is only added or deleted when the job ends OK; the DO COND condition is only added or deleted when the criteria in the ON block are satisfied.

As mentioned in the last chapter, every condition has an associated date. This can be an explicitly defined date, such as 0909, but more commonly, it is a keyword that represents an actual date. For example, the date keyword ODAT stands for the original scheduling date of the job. The meaning of original scheduling date is discussed in [Chapter 5, “AutoEdit and JCL,”](#) and is referred to in the discussion of date concepts in the *CONTROL-M for z/OS User Guide*.

A condition is actually identified by the combination of its name and date. Two conditions having the same name but different dates are considered two different conditions. In the last chapter, you wanted the run of IDJOB2 to be dependent on the run of IDJOB1 from the same day. Therefore, not only did the condition names of both jobs have to match, but so did the dates. Use of the keyword ODAT for both the IN and the OUT condition ensured that match. Had they run on different original scheduling dates, there would have been no match.

In this chapter you create another type of dependency—a dependency between different runs of the same job on different dates. IDJOB3 cannot run this month unless it successfully ran last month. The same condition name is used (IDJOB3-ENDED-OK), but with different dates for the IN condition and OUT condition.

- The date of the OUT condition is the date of the current job run; in other words, ODAT.
- The IN condition can only be satisfied by the run from the previous month. Therefore, the date of the OUT condition is PREV, a keyword that resolves to the previous original scheduling date (that is, the last run) of the job.

Displaying the Scheduling Plan for the Job

Before beginning to define IDJOB4, display the schedule plan for IDJOB3.

Option P (PLN) in the Job List screen is used to request a graphic display of a schedule for job planning.

- 30 Set the FROM DATE to the first day of the current month. Set the TO DATE to the last day of the current month, and press **Enter**.

The Job Scheduling screen is displayed.

Figure 25 Job Scheduling Screen

JOB NAME: I DJOB3		JOB SCHEDULING				DATES : 010201 - 280201			
COMMAND ==>						SCROLL==> CRSR			
02	2001	MON	TUE	WED	THU	FRI	SAT	SUN	
					01	02	03	04	
		05	06	07	08	09	10	11	
		12	13	14	15	16	17	18	
		19	20	21	22	23	24	25	
						*			
		26	27	28					
CMDS: NEXT, PREV, END								11. 32. 19	

The Job Scheduling screen displays a calendar in graphic format. The dates within the specified date range on which the job is to be scheduled are marked with an asterisk.

NOTE



If the specified date range spans multiple months, a different calendar panel is displayed for each month, beginning with the first month. You can move between panels by pressing relevant PF keys. PF keys and their associated commands are explained in the following table.

Table 8 PF Keys and Screen Navigation Commands for Job Scheduling Plan Screens

PF Keys	Associated Commands
PF07/PF19	UP (scroll backward)
PF08/PF20	DOWN (scroll forward)
PF10/PF22	PREV (where applicable)
PF11/PF23	NEXT (where applicable)

Months in the date range in which the job is not scheduled are not displayed.

You can also display the Job Scheduling screen from the Job Scheduling Definition screen. To do so, enter **PLAN** in **COMMAND** field. This is especially useful for checking your definition of a complex schedule while you are working on it.

- 31** Exit the Job Scheduling screen to display the Job List screen.

Defining JOB4

Save the definition of IDJOB3 and create IDJOB4, as explained in the following steps.

- 32** Exit the Job List screen.

The Exit option window is displayed.

Because you are saving the job scheduling definition in a table that already exists (IDGS1), you should use the **SAVE** option, not the **CREATE** option.

- 33** Specify **Y** in the **SAVE** field and leave the **CREATE** field blank.

The job scheduling definition is saved in the table. Because you bypassed the Table List screen when you entered the job scheduling definition, the entry panel is displayed.

You can now create JOB4 by specifying a new JOB name in the entry panel, as was done when you created IDJOB1 and IDJOB3, or by performing an Insert procedure in the Job List screen, as was done when you created IDJOB2.

Because JOB4 will be very similar to JOB2, you should use the Insert method. Before going to the Job List screen, however, you should first enter the Table List screen and select the table you want to use as a template for the new job.

- 34** Ensure the **TABLE** and **JOB** name fields are blank in the Entry panel, and enter the correct library name.

The Table List screen is displayed.

- 35** Enter the **SELECT** option for table IDGS1.

The Job List screen is displayed for table IDGS1. It lists jobs IDJOB1, IDJOB2, and IDJOB3.

- 36** Enter the **INSERT** option for IDJOB2.

A new job scheduling definition is displayed. It contains several values found in IDJOB2.

37 Make the following changes in the job scheduling definition.

- A** Specify **IDJOB4** in the MEMNAME field.
- B** Change the value of the GROUP field to **IDGRP2**.
- C** Add the following description in the DESC field:
AUXILIARY JOB WHEN IDJOB3 ENDS NOTOK.
- D** Delete the ALL value from the DAYS parameter.
- E** Replace the existing IN condition with **IDJOB3-END-NOTOK**, but do not change the date value ODAT.
- F** Change the OUT condition to **IDJOB4-ENDED-OK ODAT +**.

The job scheduling definition appears as follows:

Figure 26 Job Scheduling Definition for job IDJOB4

```

JOB: IDJOB4 LIB CTM. TEST. SCHEDULE TABLE: IDGS1
COMMAND ==>>> SCROLL==> CRSR
-----+-----
MEMNAME IDJOB4 MEMLIB CTM. TEST. JCL
OWNER ID TASKTYPE JOB PREVENT-NCT2 Y DFLT N
APPL GROUP IDGRP2
DESC AUXILIARY JOB WHEN IDJOB3 ENDS NOTOK
OVERLIB STAT CAL
SCHENV SYSTEM ID NJE NODE
SET VAR
CTB STEP AT NAME TYPE
DOCMEM IDJOB4 DOCLIB
=====
DAYS DCAL
AND/OR
WDAYS WCAL
MONTHS 1- Y 2- Y 3- Y 4- Y 5- Y 6- Y 7- Y 8- Y 9- Y 10- Y 11- Y 12- Y
DATES
CONFCAL RETRO N MAXWAIT 00 D-CAT
MINIMUM PDS
DEFINITION ACTIVE FROM UNTIL
=====
IN IDJOB3-END-NOTOK ODAT
CONTROL
RESOURCE
PIPE
FROM TIME + DAYS UNTIL TIME + DAYS
DUE OUT TIME + DAYS PRIORITY 00 SAC CONFIRM
TIME ZONE:
=====
OUT IDJOB4-ENDED-OK ODAT +
AUTO-ARCHIVE SYSDB MAXDAYS MAXRUNS
RETENTION: # OF DAYS TO KEEP # OF GENERATIONS TO KEEP

```


- Finally, the missing IN CONDITION (IDJOB3-ENDED-OK) is listed, unless you have an actual job that added this condition.

The problems listed above are the runtime scheduling requirements that prevent your job from running.

Runtime scheduling requirements are defined for a purpose, and it is generally not desirable to nullify them or work around them. However, sometimes it does become desirable to manually intervene to enable job submission, and CONTROL-M allows for such intervention. The following sections illustrate how you can correct the problems and enable the job run.

Adding a Missing Condition in the Why Screen

Begin by adding the missing condition. You can do this in the Why screen. Notice the A (ADD CONDITION) option listed at the bottom of the Why screen.

- 49 Tab to the CONDITION line and request option A.

A confirm window is displayed.

- 50 Confirm the request.

Figure 30 Confirmation Window for Adding Condition in Why Screen

```

----- I DJOB3  SCHEDULING ANALYSIS -----(3. ?)
COMMAND ==>
OPT DESCRIPTION                                SCROLL==>  CRSR

      TIME LIMIT      FROM  2300              UNTIL  0400
      RESOURCE        I DTAPE-DRI VE          QUANTI TY 0001
      RESOURCE        I DCPOWER
A  CONDI TI ON  I DJOB3-ENDED-OK              <-----| CONFIRM Y  DDMM 0202 |
===== >>>>>>>>>>>>>>>>>>>>>>>>>>>> END OF "WH              +-----+

```

OPTION: A ADD CONDI TI ON 14. 45. 44

The condition disappears from the Why screen, and a message appears at the top of the screen stating that the condition has been added.

IOA Conditions/Resources Screen

As already mentioned, two quantitative resources, IDTAPEDRIVE (1 unit) and IDCPOWER (5 units), are missing. You can add these missing conditions in the IOA Conditions/Resources screen.

51 Enter =4 in the COMMAND field of the Why screen.

The IOA Conditions/Resources screen is displayed.

Figure 31 IOA Conditions/Resources Screen

```

----- IOA CONDITIONS/RESOURCES -----(4)
COMMAND ==>
PREFIX ==>
COND Y CONTROL Y RES Y STAT Y
OPT TYPE CONDITION/RESOURCE IOA ID USE QUANTITY MAX *P RBA DATE
CONTROL CONTROLM 01 E 0003 0003 (00000)
RESOURCE TAPEP 0003 0003
RESOURCE CPU1 0098 0100
RESOURCE CPU2 0197 0200
RESOURCE TAPEP 01 Q 0002 (00091)
RESOURCE CPU1 01 Q 0002 (00091)
RESOURCE CPU2 01 Q 0003 (00092)
RESOURCE TAPEP 01 R 0002 1 (00093)
COND BR-BRI VPCC-ENDED-OK 0202
COND BR-BRCC0001-ENDED-OK 0202
COND BR-BRCC0002-ENDED-OK 0202
COND BR-BRCC0003-ENDED-OK 0202
COND BR-BRCCIND-ENDED-OK 0202
COND BR-BRUPDT02-ENDED-OK 0202
COND BR-BRREP001-ENDED-OK 0202
COND BR-BRREP002-ENDED-OK 0202
COND GL-GLI NP001-ENDED-OK 0202
COND EBD-APPL-STARTED 0202
COND CICS-PROD-ISIS-UP STAT
OPTI ONS: D DELETE C CHANGE COMMANDS: ADD 14.07.08

```

The IOA Conditions/Resources screen displays the list of existing Control and Quantitative resources, which are stored in the CONTROL-M Resources file, and the list of existing conditions stored in the IOA Conditions file.

Filtering the IOA Conditions/Resources Screen

When the list of conditions and resources in the IOA Conditions/Resources screen is very long, say ten or twenty or even fifty pages, it can be very useful to filter the display.

You can filter the display based on prefix and/or based on type of resource or condition. You can now filter the display so that only conditions and resources beginning with your ID are showing.

Zoom Screen

The Zoom screen is very similar to the Job Scheduling Definition screen. Instead of displaying the permanently defined job scheduling definition, however, the Zoom screen displays the job scheduling definition of the requested job order.

In the Zoom screen, you can change or delete TIME FROM and TIME UNTIL criteria for a job order. You can now display the Zoom screen, which is requested from the Active Environment screen.

58 Exit the Why screen.

The Active Environment screen is displayed. Note the Z (Zoom) option at the bottom of the Active Environment screen. This option displays the Zoom screen.

You will recall from [Chapter 2, “Scheduling Definition and Manual Intervention,”](#) that before you could delete a job from the Active Environment screen, you had to place the job in Held status. To make changes through the Zoom screen, you must first hold the job in the Active Environment screen, and then enter the Zoom screen.

59 Enter option H for IDJOB3. Press Enter several times until the status of the job is changed to Held Wait Schedule.

60 Enter option Z for IDJOB3.

The Zoom screen is displayed.

Figure 33 Zoom Screen

```

----- CONTROL-M ZOOM SCREEN -----(3. Z)
COMMAND ==> SCROLL==> CRSR
+-----+
MEMNAME I DJOB3      MEMLIB  CTM. TEST. JCL
OWNER   ID          TASKTYPE JOB          PREVENT-NCT2  DFLT- N
SCHDTAB I DGS1      SCHDLIB CTM. TEST. SCHEDULE
APPL                                         GROUP
OVERLIB
SCHENV          SYSTEM ID          STAT CAL
JOBNAME        JOBI D          ODATE O20201 ORDERID 00100  MAXWAIT 05
RESTART DECISION-FROM          TO          CONFIRM N
DESC  FISCAL RESULTS ON LAST FRIDAY OF MONTH
SET VAR
CTB STEP AT          NAME          TYPE
DOCMEM I DJOB3      DOCLIB
-----+-----+
IN          I DJOB3-ENDED-OK      0202
CONTROL
RESOURCE I DTAPE-DRIVE          0001      I DCPOWER          0005
PIPE
FROM TIME 2300 +      DAYS      UNTIL TIME 0400 +      DAYS
DUE OUT TIME 1159 +      DAYS      PRIORITY *5 SAC      CONFIRM N
TIME ZONE:
COMMANDS:  SAVE  DOC  NOTE          12. 54. 56

```

There are differences between the permanent definition displayed in the Job Scheduling Definition screen, and the job order definition displayed in the Zoom screen. For example, the definition in the Zoom screen does not display Basic scheduling parameters—these parameters are irrelevant once the job order is in the Active Environment. And the Zoom screen displays parameters such as SEARCH COUNTER, which are not found in the job scheduling definition, but which are relevant to the job order processing and have values set by CONTROL-M.

The runtime scheduling parameters you defined for IDJOB3 appear in the Zoom screen.

```

-----+-----+
IN          I DJOB3-ENDED-OK      0202
RESOURCE I DTAPE-DRIVE          0001      I DCPOWER          0005
PIPE
FROM TIME 2300 +      DAYS      UNTIL TIME 0400 +      DAYS
DUE OUT TIME 1159 +      DAYS      PRIORITY *5 SAC      CONFIRM N
TIME ZONE:          WAIT FOR ODATE:
CPU-ID      NODE NAME          NJE      SEARCH COUNTER 00000
-----+-----+

```

Of the runtime scheduling criteria you defined, the TIME FROM and TIME UNTIL range remains the only problem. You can resolve this by simply changing the TIME values in the Zoom screen.

- 61 Change TIME FROM to a value that is an hour earlier than your current system time. Change the UNTIL value to a time that is several hours after the current system time.

At this point, it is important that you note the following:

- Any change to a job order in the Zoom screen changes the value for that job order only. The permanent definition that you defined in the Job Scheduling Definition screen remains unchanged.
- Instead of using the CONTROL-M Why and IOA Conditions/Resources screens to satisfy the other runtime parameters, you could have deleted those parameters from the job order in the Zoom screen.
- The most important thing to remember is that in actual production circumstances, you do not merely delete parameter values that “get in the way,” and you do not intervene in the job control process unless you are fully aware of the potential impact of that intervention.

These exercises are designed to show you the workings of scheduling parameters and to show you how you can intervene if absolutely necessary. But runtime criteria are defined for a reason, and unless you are certain of what you are doing, intervention can do much more harm than good. For example, intervening to cause the submission of a job before its predecessor job has run can potentially do great damage.

Therefore, intervention in job processing should be left only to those who are specifically authorized and who have the knowledge necessary to intervene appropriately.

You can now exit the Zoom screen and see the result.

- 62 Enter SAVE in the COMMAND field of the Zoom screen. The Active environment screen is displayed.

Freeing a Held Job

Because you held the job before entering the Zoom screen, the job cannot continue processing until you free it. You do this by entering option F (Free) for the job.

- 63 Enter option F for the job order for IDJOB3. A message at the top of the screen indicates that the freeing of the job is performed. The status of the job changes to Requested Free Wait Schedule.

```

CTM659I FREE OF TASK IDJOB3 GROUP " " ODATE 020201 PERFORMED <D> - (3)
COMMAND ==> SCROLL ==> CRSR
O Name Owner Odate Jobname JobID Typ ----- Status -----
IDJOB3 ID 020201 JOB Requested Free Wait Schedule
===== >>>>>>>>>> Bottom of Jobs List <<<<<<<<<<<< =====

```

64 Periodically refresh the Active Environment screen by pressing **Enter**.

During this process, you see the following actions:

- IDJOB3 changes to Wait Schedule status, and then goes through the regular submission process. The job ends NOTOK (with a condition code of C0008).
- The Shout message you defined is sent to your terminal. The message is similar to the following:

```

CTM- FRI DAY FISCAL RESULTS RUN FAILED 02.02 14:12
CN(INTERNAL)

```

- As a result of the condition code from IDJOB3, job IDJOB4 is forced and is displayed in the Active Environment screen. Job IDJOB4 ends OK.

```

Filter: IDGS ----- CONTROL-M Active Environment ----- UP <D> - (3)
COMMAND ==> SCROLL ==> CRSR
O Name Owner Odate Jobname JobID Typ ----- Status -----
IDJOB3 ID 020201 M21 /31866 JOB Ended- Not "OK" Due to CC
IDJOB4 ID 020201 M21 /31867 JOB Ended "OK"
===== >>>>>>>>>> Bottom of Jobs List <<<<<<<<<<<< =====

```

If desired, you can check the results of the job runs in the Log file. As you did in the last chapter, you can enter option **L** for either or both jobs.

Alternatively, you can enter **=5** in the COMMAND line to request option 5 of the IOA Primary Option menu. This option displays the IOA Log screen. Once in the IOA Log screen, you can define a filter to filter the job display in much the same way that you did in the Active Environment screen.

65 Exit the Online facility.

This completes the steps in this chapter of the *CONTROL-M for z/OS Getting Started Guide*.

Review

In this chapter you

- looked closely at the sections in the Job Scheduling Definition screen, and learned about the following parameters that were not discussed in the last chapter
 - WDAY
 - MONTH
 - (DATE)
 - CONTROL
 - RESOURCE
 - TIME FROM / UNTIL
 - PRIORITY
 - ON
 - DO COND
 - DO SHOUT
 - DO FORCEJOB
- used the following options of the Job List screen
 - P (Display the Job Scheduling Plan screen)
 - F (Force a job order)
- used the following options of the Active Environment screen
 - H (Hold)
 - F (Free)
 - ? (Display the Why screen)
 - Z (Display the Zoom screen)
- viewed the scheduling plan of a job in the Job Scheduling Plan screen
- used the Why screen to see why a job didn't run, and to add a missing condition
- filtered the display of the IOA Conditions/Resources screen, and used the screen to add Quantitative resources
- used the Zoom screen to alter the runtime parameters of a job order
- learned about the difference between ordering and forcing a job
- learned more prerequisite conditions and their associated dates, including the PREV date

Recommended Reading

Before continuing with the next chapter, it is recommended that you read the following in the *CONTROL-M for z/OS User Guide*.

In Chapter 2

Description of the Job Scheduling Plan screen, Why screen, Zoom screen and IOA Conditions/Resources screen

In Chapter 3

Detailed parameter descriptions of the parameters you have used (mentioned above, under “Review”)



NOTE

The descriptions of the ON and WDAY parameters are quite detailed. Read those descriptions for a general understanding of parameter usage and possible parameter formats, without trying to memorize all the detail. For now, you can skip the information related to subparameter WCAL in the WDAY parameter description.

Restarts under CONTROL-M/Restart

This chapter includes the following topics:

Overview	92
Preparations	92
Defining Restart in the Job Scheduling Definition	93
Editing JCL from the Active Environment	99
Restarting the Job	100
Review	104
Recommended Reading	104

Overview

In the last chapter, you used a DO FORCEJOB statement in an ON/DO block to force a “remedial” job following a job failure. However, rather than run a remedial job following job failure, it is more likely that you will want to correct the problem and then restart the job that failed.

In this chapter, you will learn to use CONTROL-M/Restart to perform job restarts when they become necessary. Before you do, however, you should be clear about the difference between a job rerun and a job restart.

Job rerun is the re-execution of a scheduled job, starting from the beginning. For example, if a job fails, the entire job can be rerun. At best, rerunning a job can waste processing time on already successfully completed job steps; and unless certain precautions are taken, if successful job steps from the prior run performed updates before the job failed, rerunning the job can create problematic results by repeating those updates.

Job restart is the re-execution of a job beginning at a particular step. In general, the results of successful job steps before the failure are utilized, and re-execution continues from the end of the last successful step.

CONTROL-M/Restart automates restart by identifying the step at which to initiate a job restart, and by performing necessary tasks to ensure that job restart is error-free.

Two separate processes are required for CONTROL-M/Restart to restart under CONTROL-M:

- Defining the restart parameters in the CONTROL-M job scheduling definition appropriately, so that restart can be performed if it becomes necessary.
- Activating the restart process when restart becomes necessary.

In this chapter, you will define restart parameters in the job scheduling definition, and then, following job failure and correction of the problem, see and involve yourself in the process of restart.

Preparations

For this chapter, you will create a new job and its job scheduling definitions. You will use the name IDJOB5, and the same library and table that you used in the preceding chapter.

To create the JCL for IDJOB5, begin with a copy of the TESTUTIL JCL, such as the JCL you used for IDJOB3. Recall that for job IDJOB3 this JCL had one step, which you defined to end with a condition code of C0008. For IDJOB5, you should make the following changes:

1. Change the step so that it ends with a condition code of C0000 (so the step ends OK).
2. Copy the changed step and its accompanying DD statements four times, so that you have five steps in the job. Name those steps S1, S2, S3, S4, and S5, respectively.
3. Change the third step (S3) so that it ends with a condition code of C0008.

Your job should now have five steps, four of which end okay, but one, the third, that ends with a condition code of C0008.

You can now continue with the first part of these exercises; the creation of the job scheduling definition.

Defining Restart in the Job Scheduling Definition

- 1 Enter the IOA Online Facility and open a job scheduling definition for job IDJOB5.
- 2 Ensure that the following values are part of the job scheduling definition:
 - In the MEMNAME field, specify **IDJOB5**. Fill in the appropriate JCL library name in the MEMLIB field.
 - In the GROUP field, specify **IDGRP3**.
 - In the DESC field, specify **RESTART JOB UNDER CTM/RESTART**.
 - In the DAYS field, specify **ALL**, and specify **Y** in all the MONTHS fields. These are the only Basic Scheduling parameters you should define.
 - Do not defined any Runtime Scheduling parameters.
 - In the OUT fields, define the condition

```
I DJOB5-ENDED-OK   ODAT   +
```

You are now ready to define the ON and DO statements. These will include your restart parameters.

- 3 In the ON block, specify **ANYSTEP** as the program step (PGMST) value. Specify **>C0004** as the codes value.

The meaning of the >C0004 value should be clear. Generally, a step is considered to have ended OK if the routine returns a code of C0004 or lower. Therefore, a CODES value of >C0004 instructs CONTROL-M to perform the accompanying DO statements if the job ended NOTOK.

In the preceding chapter, and again in this step, you specified ANYSTEP as the PGMST step values. These ON *step* values should be examined more closely.

ON Steps

The ON *step* fields identify the possible steps against which CONTROL-M will check for the specified CODES values.

Two types of step values can be specified:

- Program Step (PGMST) value
- Procedure Step (PROCST) value

You can specify either or both types of values, but you must specify at least one value if you use an ON block.

Valid step values can be any of the following:

- Literal value (for example, S3, if this is a step name)
- Keyword value that represents a step

Valid keyword values are

— ANYSTEP, which is valid only as a PGMST value

DO statements are performed if the CODES criteria are satisfied for any program step.

— +EVERY

DO statements are performed only if the CODES criteria are satisfied for all steps, program and/or procedure, depending on the definition.

- Step Range Name, which is valid only for a PGMST value

You can define a range of steps in the STEP RANGE statement, immediately above the ON statement. You must assign a name to this step range.

You can then specify this step range name, preceded by an asterisk, as the step value in the ON statement. The asterisk prefix in the ON statement instructs CONTROL-M to check the program step range defined in the STEP RANGE field, rather than looking for an actual program step by that name.

DO statements are performed if the CODES criteria are satisfied for any program step in the step range.

- 4 Define the following DO COND statement as the first DO statement in the ON block:

IDJOB5-END-NOTOK ODAT +

You can now to define a Shout statement to the operator. However, because this is only an exercise and you do not want the shout to actually go to the operator, you will instead send the shout to yourself.

- 5 Define a DO SHOUT statement as the next DO statement in the ON block. The target location, or TO value, defaults to your user ID. Specify the message:

IDJOB5 RUN FAILED. CORRECT PROBLEM AND CONFIRM RESTART.

You are now ready to define your restart parameters. Two statements are generally used in combination, to define a restart:

- DO IFRERUN
- DO RERUN

The DO IFRERUN statement is used to indicate that a restart is desired. It defines parameters to be used for restart, such as FROM step and TO step. This statement tells CONTROL-M that if the job is rerun, hence the name DO IFRERUN, it should be rerun in accordance with the restart values specified in the DO IFRERUN statement.

The DO RERUN parameter merely instructs CONTROL-M to run the job again. If you wanted a rerun without a restart, you would specify only the DO RERUN parameter, and the job would be rerun from the beginning.

The DO IFRERUN statement precedes the DO RERUN statement.

- 6 Enter **IFRERUN** in the blank DO statement. The DO IFRERUN statement is opened.

Note that this statement contains FROM, TO, and CONFIRM fields. The default value of the CONFIRM field should be N (No).

- 7 Set the CONFIRM value to Y (Yes).



NOTE

The CONFIRM field of the DO IFRERUN statement is similar in meaning to the CONFIRM runtime scheduling parameter that you used in [Chapter 1, “Introduction to CONTROL-M.”](#) However, it applies only to restarted jobs, whereas the CONFIRM runtime scheduling parameter applies to all job runs.

You can now take a look at the FROM and TO fields.

The FROM field indicates the step from which the restart should begin, and the TO field indicates the step to which the restart should continue.

Both the FROM and the TO fields allow specification of a program step, to the left of the period, and/or a procedure step, to the right of the period. You can specify either or both values.

- A TO step value can only be a literal value, such as S3, if this is a step name, or blank. If no TO value is specified, job processing continues to the end.
- The FROM field allows specification of either a literal value or a keyword that represents a step. Some of the valid keyword values for the FROM step are listed in the following table:

Table 9 DO IFRERUN: Selected FROM Keyword values

Keyword	Description
\$FIRST	First step of the job.
\$ABEND	Step of the job that ended NOTOK due to system abend, user abend, condition code C2000 (PL/1 abend), or JFAIL (job failed on JCL error). \$ABEND is a subset of \$EXERR, described below.
\$FIRST.\$ABEND	First step of the abended procedure.
\$EXERR	Job step that ended with any error, including an abend, or that ended with a condition code that is redefined using the ON and DO statements, as ENDED NOTOK

8 Specify **\$EXERR** in the FROM field.

9 Enter **RERUN** in the blank DO statement. This statement has no subparameter values.

Before exiting the job scheduling definition, return and define one more OUT condition.

10 In the OUT condition line, add the following condition as the second condition on that line:

```
I DJOB5-END-NOTOK ODAT -
```


The bottom half of the window deals with restart information. In this exercise, you will only look at the first line, which tells you from which step, and to which step, the restart will be performed.

- The FROM value is S3. This makes sense because steps S1 and S2 ended successfully.
- The TO step is blank, which means that once restart begins, it will continue till the end of the job. Consider the following:
 - If you do not want the steps after the restart step to run again, you can specify restart step S3 as the TO step.
 - If you defined the JCL so that steps after the failed step do not run, and you want them to run following the restart, you should leave the TO value blank.

At this point, all you need to do is enter Y in the CONFIRM field, and the job will restart. However, you should not take that action at this time.

There might be instances in which you want the job to restart from a different step than the one determined by CONTROL-M/Restart as the logical restart step. It is possible to change the FROM and TO steps in the Confirm Restart window. To facilitate this change, you can display the list of steps in the job.

Notice that ? is a valid value for the With Restart field. Entering ? displays the list of steps.

- 22 Enter ? in the **With Restart** field. The Restart Step List window is opened over the Restart Window.

Review

In this chapter you

- defined a restart in your job scheduling definition using parameters DO IFRERUN and DO RERUN
- learned valid restart step keyword values and specified that the job should restart from step \$EXERR
- entered the JCL of the failed job from the Active Environment screen by entering the J (JCL) option, and corrected the JCL
- confirmed a rerun/restart request (Option C) for the failed job in the Active Environment screen, and in the process you displayed the Confirm Restart Window and Restart Step List window, in which you confirmed the restart
- checked the log of the job following the failed run, and again following the restart

Recommended Reading

Before continuing with the next chapter, it is recommended that you read the following:

- In the *CONTROL-M/Restart User Guide*
 - all of Chapter 1.
 - in Chapter 2, the online material related to the Scheduling Definition facility and the Tracking and Control facility.
- In the *CONTROL-M for z/OS User Guide*.
 - In Chapter 2, the description of the CONTROL-M/Restart information related to the Confirm Restart window and Rerun/Restart window, and the Restart Step List window.
 - In Chapter 3, detailed parameter descriptions of parameters DO IFRERUN and DO RERUN.

Group Scheduling

This chapter includes the following topics:

Overview	106
Preparations	106
Creating a Group Scheduling Table	107
Group Entity	108
Planned Logic of the Job Scheduling Definitions	115
Jobs Scheduling Definitions in the Group Table	115
Saving the Table	126
Points to Remember Concerning Group Scheduling	126
Group Entity Ordering and Job Ordering	126
Advantages of Group Scheduling	130
Review	131
Recommended Reading	132

Overview

When defining job scheduling definitions, specification of a group name is optional. Groups are user constructs. You decide if and what jobs should be grouped together, and the name to be assigned to that group.

It is good practice to group together related job scheduling definitions. This is not a requirement—related jobs can belong to different groups, or not belong to any groups—but it can be useful. For example, if related jobs belong to the same group, you can use the group name to filter the Active Environment screen so that only the jobs in that group are displayed. But groups can provide far greater functionality than just ease of filtering a display.

The major advantage of defining groups is that they enable you to define basic scheduling criteria, runtime scheduling criteria and post processing criteria that apply to the group of jobs as a whole.

Preparations

In the exercises in the previous chapters of this guide, you created the JCL for five jobs, IDJOB1 through IDJOB5. To simplify preparation for this chapter, you will utilize all five of these JCLs.

In addition, you will utilize one of the job scheduling definitions that you created in [Chapter 1, “Introduction to CONTROL-M,”](#) the job scheduling definition for IDJOB1, in table IDGS1. For the remaining jobs, you will create new job scheduling definitions in a new table (IDGS2), as part of the exercises.

Before continuing with the exercises, you should perform the following preparatory steps:

- In the JCL of IDJOB5, change the JCL of step S5 so that it sets a return code of C0008 instead of C0000.
- Ensure the following, which should be the case if you completed all of the steps in the previous chapters of this guide:
 - The JCL for step S1 in IDJOB3 sets a return code of C0000, and not C0008.
 - All conditions from previous exercises in this guide do not appear in, or are deleted from, the IOA Conditions/Resources screen.
 - All job orders from previous exercises in this guide do not appear in, or are deleted from, the Active Environment screen.

You are now ready to look at group scheduling. For purposes of these exercises, you should assume that the new table contains jobs that must run at end of week and at end of month.

Creating a Group Scheduling Table

- 1 Enter the IOA Online Facility and display the Scheduling Definition Facility entry panel.

```

----- CONTROL-M SCHEDULING DEFINITION FACILITY - ENTRY PANEL -----(2)
COMMAND ===>

SPECIFY LIBRARY, SCHEDULING TABLE, JOB

LIBRARY ===> CTM.TEST.SCHEDULE
TABLE ===> (Blank for table selection list)
JOB ===> (Blank for job selection list)

TYPE OF TABLE ===> ( J Job - default
                     G Group - for new tables only)

SHOW JOB DOCUMENTATION ===> N (Y/N)
AUTO-SAVE DOCUMENTATION ===> N (Y/N)

USE THE COMMAND SHPF TO SEE PFK ASSIGNMENT
23.00.04

```

In this chapter, you will create a new scheduling table called IDGS2; in the previous chapters of this guide you used a scheduling table called IDGS1.

Notice the field TYPE OF TABLE. Valid values for this field are J (Job) and G (Group). In previous chapters of this guide, you ignored this field because the default value (Job) served your purpose when you worked with table IDGS1. Now, however, you want to define the new table, IDGS2, to be a Group table.

The main difference between a Job-type table and a Group-type table is that a Job-type table does not contain a mechanism for defining parameters that apply to the group as a whole, but a Group-type table does.

In the previous chapters, whenever you defined a job scheduling definition in table IDGS1, the parameters in the job scheduling definition could only be applied to that job. For example, a TIME FROM value, or a DO SHOUT value, in the job scheduling definition for IDJOB3, applied only to IDJOB3. If you wanted to apply the same parameters to IDJOB4, you had to define the parameters in the job scheduling definition of IDJBO4.

Group tables, however, have a special kind of scheduling definition, called a Group entity, that is used to apply parameters to all jobs in the group. Each Group table has one, and only one, Group entity, as well as its job scheduling definitions. When you first request creation of a Group scheduling table from the Scheduling Definition entry panel, the Group entity definition screen is displayed.

Group Entity

You can now request creation of Group scheduling table IDGS2, and take a look at the Group entity.

- In the Entry Panel, enter **IDGS2** in the TABLE field and **G** in the TYPE OF TABLE field.

```

SPECIFY LIBRARY, SCHEDULING TABLE, JOB

LIBRARY ====> CTM. TEST. SCHEDULE
TABLE   ====> IDGS2           (Blank for table selection list)
JOB     ====>                 (Blank for job selection list)

TYPE OF TABLE      ====> G   ( J Job - default
                               G Group - for new tables only)
    
```

The Group entity scheduling definition is displayed.

Figure 37 Group Entity Scheduling Definition

```

GRP                                CTM. TEST. SCHEDULE(IDGS2)
COMMAND ====>                                SCROLL====> CRSR
+-----+-----+-----+-----+-----+-----+-----+-----+
GROUP                                MEMNAME
OWNER  ID
APPL
DESC
ADJUST CONDITIONS N                GRP MAXWAIT 05
SET VAR
DOCMEM                                DOCLIB
=====
SCHEDULE TAG
DAYS                                DCAL
                                AND/OR
WDAYS                                WCAL
MONTHS  1- Y 2- Y 3- Y 4- Y 5- Y 6- Y 7- Y 8- Y 9- Y 10- Y 11- Y 12- Y
DATES
CONFICAL                                SHIFT                RETRO N MAXWAIT 05
SCHEDULE TAG ACTIVE FROM                UNTIL
=====
IN
CONTROL
=====
COMMANDS: EDIT, DOC, PLAN, JOBSTAT                                11. 51. 58
    
```

Although the Group entity scheduling definition is similar to the job scheduling definition, there are differences. The first thing to note is that the term GRP in the upper left corner identifies the definition as a Group entity.

The next thing to note is that the first field in the definition, under the COMMAND line, is the GROUP field, followed by the MEMNAME field on the same line. Note also that there is no MEMLIB field.

In a regular job scheduling definition, the MEMNAME parameter contains the name of the JCL member of the job. However, a Group entity does not have a JCL member. Therefore, any value can be specified in the MEMNAME field. But keep in mind that whatever name you specify in the MEMNAME field automatically gets placed in the DOCMEM field.

- 3 In the GROUP field, specify the name **IDGRP4**, which is the name of our group. In the MEMNAME field, specify the value **GRPSCHD**.
- 4 In the DESC field, specify **GROUP ENTITY FOR TABLE IDGS2**. The entries in the General parameters section appear as shown below:

```

+-----+
GROUP  IDGRP4                MEMNAME  GRPSCHD
OWNER  ID
APPL
DESC   GROUP ENTITY FOR TABLE IDGS2
ADJUST CONDITIONS N        GRP MAXWAIT 05
SET VAR
DOCMEM GRPSCHD            DOCLIB
=====

```

Schedule Tags and Other Basic Scheduling Parameters

You can now look at the Basic Scheduling parameters section. The first field to note is the SCHEDULE TAG field. This field only appears in definitions in Group tables.

```

=====
SCHEDULE TAG
DAYS                                DCAL
                                      AND/OR
WDAYS                                WCAL
MONTHS 1- Y 2- Y 3- Y 4- Y 5- Y 6- Y 7- Y 8- Y 9- Y 10- Y 11- Y 12- Y
DATES
CONF CAL          SHIF T          RETRO N MAXWAIT 05
SCHEDULE TAG ACTIVE FROM          UNTIL
=====

```

- 5 In the SCHEDULE TAG field, specify the name **EOW**, which stands for END-OF-WEEK. In the WDAYS field, specify the value **0**, which means the last day of the week.

```

=====
SCHEDULE TAG EOW
DAYS                                DCAL
                                      AND/OR

WDAYS  0                            WCAL
MONTHS 1- Y 2- Y 3- Y 4- Y 5- Y 6- Y 7- Y 8- Y 9- Y 10- Y 11- Y 12- Y
DATES
CONFCAL          SHI FT          RETRO N MAXWAIT 05
SCHEDULE TAG ACTIVE FROM          UNTI L
=====
SCHEDULE TAG
DAYS                                DCAL
                                      AND/OR

WDAYS                                WCAL
MONTHS 1- Y 2- Y 3- Y 4- Y 5- Y 6- Y 7- Y 8- Y 9- Y 10- Y 11- Y 12- Y
DATES
CONFCAL          SHI FT          RETRO N MAXWAIT 05
SCHEDULE TAG ACTIVE FROM          UNTI L
=====

```

Notice that a new, blank set of Basic Scheduling criteria opens up.

Thus far, in each of your job scheduling definitions, you have been limited to defining only one set of Basic scheduling parameters. This is because a job scheduling definition can have only one occurrence of the Basic Scheduling parameter section.

Group entities, on the other hand, can have multiple occurrences of the Basic scheduling parameters section. Once you have defined a set of Basic Scheduling parameters, a new, blank section of Basic Scheduling parameters opens to enable you to define another set.

You distinguish between the sets of Basic Scheduling parameters in the Group entity by assigning a name to each set in the SCHEDULE TAG field.

In the “Preparations” section above, you determined that the jobs in this table will run at the end of the week and at the end of the month. You have just defined the basic scheduling criteria for the end of the week run. You can now define the basic scheduling criteria for the end of the month run.

- 6 In the empty SCHEDULE TAG field, specify the name **EOM**, which stands for END-OF-MONTH. In the DAYS field, specify the value **L1**, which means the last day of the month.

In an actual operational environment, these sets of Basic Scheduling criteria would be enough to achieve your goals. However, because you want to be able to perform these exercises on any day of the month, you will define one more set of Basic scheduling criteria for purposes of this guide only.

- 7 In the blank SCHEDULE TAG field, specify the name **EXERCISES**. In the DAYS field, specify the value **ALL**.

Your entries in the Basic Scheduling parameter section are displayed as shown in [Figure 38](#).

Figure 38 Basic Scheduling Parameter Section

```

=====
SCHEDULE TAG EOW
DAYS                                     DCAL
                                         AND/OR
WDAYS 0                                 WCAL
MONTHS 1- Y 2- Y 3- Y 4- Y 5- Y 6- Y 7- Y 8- Y 9- Y 10- Y 11- Y 12- Y
DATES
CONFCAL          SHI FT          RETRO N MAXWAI T 05
SCHEDULE TAG ACTI VE FROM          UNTI L
=====
SCHEDULE TAG EOM
DAYS  L1                                 DCAL
                                         AND/OR
WDAYS                                     WCAL
MONTHS 1- Y 2- Y 3- Y 4- Y 5- Y 6- Y 7- Y 8- Y 9- Y 10- Y 11- Y 12- Y
DATES
CONFCAL          SHI FT          RETRO N MAXWAI T 05
SCHEDULE TAG ACTI VE FROM          UNTI L
=====
SCHEDULE TAG EXERCISES
DAYS  ALL                                 DCAL
                                         AND/OR
WDAYS                                     WCAL
MONTHS 1- Y 2- Y 3- Y 4- Y 5- Y 6- Y 7- Y 8- Y 9- Y 10- Y 11- Y 12- Y
DATES
CONFCAL          SHI FT          RETRO N MAXWAI T 05
SCHEDULE TAG ACTI VE FROM          UNTI L
=====
SCHEDULE TAG
DAYS                                     DCAL
                                         AND/OR
WDAYS                                     WCAL
MONTHS 1- Y 2- Y 3- Y 4- Y 5- Y 6- Y 7- Y 8- Y 9- Y 10- Y 11- Y 12- Y
DATES
CONFCAL          SHI FT          RETRO N MAXWAI T 05
SCHEDULE TAG ACTI VE FROM          UNTI L
=====

```

Runtime Scheduling Parameters

All Runtime Scheduling fields found in the Group entity are also found in the job scheduling definition.

In the “Preparations” section above, you determined that you would continue to use the job scheduling definition of IDJOB1 in table IDGS1.

Assume that job IDJOB1, in table IDGS1, must end OK before any of the jobs in table IDGS2 can run. You will now add the condition necessary to establish this job dependency.

- 8 In the IN field, specify the condition **IDJOB1-ENDED-OK ODAT**. Your entry in the Runtime Scheduling parameter section is displayed.

IN	IDJOB1-ENDED-OK	ODAT
CONTROL		
FROM TIME	+ DAYS	UNTIL TIME + DAYS

On Group End and Other Post-Processing Parameters

Like the job scheduling definition, the Group entity contains OUT and SHOUT WHEN statements, and DO blocks. However, instead of regular ON blocks, the Group entity contains ON GROUP-END blocks.

Unlike the ON block, which applies to the individual job, the ON GROUP-END block applies to the jobs in the group as a whole. Valid values for the ON GROUP-END field are as follows:

Table 10 Valid Values For the ON GROUP-END Statement

Value	Description
OK	Subsequent DO actions are performed for each job in the group only if the end status of the Group Entity is OK, that is, all scheduled jobs in the table ended OK.
NOTOK	Subsequent DO actions are performed for each job in the group if the end status of the Group Entity is NOTOK, that is, at least one job in the group ended NOTOK.

Not all DO statements that are valid in the job scheduling definition are valid in the Group entity. The DO statements listed below can be specified in the Group entity. Several of these DO statements have not been described in this guide, but they are all described in the *CONTROL-M for z/OS User Guide*:

- DO COND
- DO OK
- DO MAIL
- DO FORCEJOB
- DO SET
- DO NOTOK
- DO SHOUT

You can now fill in the Post-processing parameters of the Group entity.

- 9 In the OUT field, specify the condition **IDGS2-ENDED-OK ODAT +**.

Logic of the Group Entity Scheduling Definition

The following points about the logic of your Group entity should be noted:

- The Group entity is used to help determine and control the processing of the jobs in the table.
 - You defined the following three sets of Basic Scheduling Criteria:
 - EOW criteria result in jobs being scheduled on the last day of the of week.
 - EOM criteria result in jobs being scheduled on the last day of the month.
 - EXERCISES criteria result in jobs being scheduled every day, for purposes of the exercises in this guide.

You will soon see how these sets of scheduling criteria apply to the job scheduling definitions in this table.

- You defined an IN condition: IDJOB1-ENDED-OK. This condition is set when job IDJOB1 in table IDGS1 ends successfully. This means that successful completion of IDJOB1 is a prerequisite to the submission of the jobs in table IDGS2 that are scheduled according to Basic scheduling criteria in the Group entity.
- You defined an OUT condition that is set only when all scheduled jobs in table IDGS2 end OK. This condition can be used as an IN condition to some other job that requires successful completion of the jobs in table IDGS2 as a prerequisite.
- You defined two sets of ON GROUP-END/DO parameters:
 - The first set applies if all the scheduled jobs in the table end OK. It shouts a message to that effect,
 - The second set applies if any of the scheduled jobs in the table ends NOTOK. It shouts a message to that effect; and it sets an appropriate condition that can, for example, be used to run an auxiliary job.

You are about to exit the Group entity and create the job scheduling definitions in table IDGS2. However, before doing so, outline the logic you would like to see in the job scheduling definitions.

Planned Logic of the Job Scheduling Definitions

As mentioned above, you want successful completion of job IDJOB1 to be a prerequisite to the running of the jobs in table IDGS2. In addition to this, the following logic should apply:

- Table IDGS2 will contain jobs IDJOB2, IDJOB3, IDJOB4, and IDJOB5.
- IDJOB2 is a prerequisite to IDJOB3, and IDJOB4 is a prerequisite to IDJOB5. However, IDJOB3 is *NOT* a prerequisite to IDJOB4.
- IDJOB2 should be run both at end of week and end of month, and during the exercise.
- IDJOB3 should be run following IDJOB2, at end of week or month, and during the exercise, provided that the day is a weekday, that is, on Monday, Tuesday, Wednesday, Thursday, or Friday.
- IDJOB4 should be run both at end of week and at end of month, but should also run on several other days of the month: the 8th, the 14th, and, so the exercise will work, the day that you perform this exercise.
- IDJOB5 should run following IDJOB4, but only at end of week and end of month, and during the exercise. It should not run the other days specified.

Jobs Scheduling Definitions in the Group Table

You should now exit the Group entity and create the job scheduling definitions in table IDGS2.

- 15 Exit the Group entity. Upon exiting the Group entity the first time, a blank Job Scheduling Definition screen is displayed.

As with all job scheduling definitions, **JOB:** is displayed in the upper left corner. Notice, however, that the **GROUP** field already has a value, which is **IDGRP4**. The name specified in the **GROUP** field of the Group entity is automatically assigned as the **GROUP** name to all the jobs in the table.

```

JOB:          LIB CTM. TEST. SCHEDULE          TABLE:  IDGS2
COMMAND ==>>          SCROLL==>>  CRSR
-----+-----
MEMNAME          MEMLIB
OWNER  ID          TASKTYPE JOB    PREVENT-NCT2  DFLT  N
APPL          GROUP IDGRP4
DESC
OVERLIB          STAT CAL
SCHENV          SYSTEM ID          NJE NODE
SET VAR
CTB STEP AT          NAME          TYPE
DOCMEM          DOCLIB
=====
SCHEDULE TAG
RELATIONSHIP (AND/OR) 0
DAYS          DCAL
          AND/OR
WDAYS          WCAL
MONTHS  1- Y 2- Y 3- Y 4- Y 5- Y 6- Y 7- Y 8- Y 9- Y 10- Y 11- Y 12- Y
DATES
CONFICAL          SHIFT          RETRO N MAXWAIT 05  D-CAT
MINIMUM          PDS
DEFINITION ACTIVE FROM          UNTIL
COMMANDS:  EDIT, DOC, PLAN, JOBSTAT          07. 38. 42
    
```

Notice that job scheduling definitions in Group tables contain two Basic Scheduling parameter fields not found in regular tables: **SCHEDULE TAG** and **RELATIONSHIP**.

The **SCHEDULE TAG** field enables you to assign to the job scheduling definition any desired sets of basic scheduling criteria that you defined in the Group entity.

- 16** Specify **IDJOB2** in the **MEMNAME** field and the name of the JCL library you are using in the **MEMLIB** field.
- 17** Specify **JOB2** in the **DESC** field. Your entries in the General parameters section are displayed as follows:

```

-----+-----
MEMNAME IDJOB2          MEMLIB  CTM. TEST. JCL
OWNER  ID          TASKTYPE JOB    PREVENT-NCT2  DFLT  N
APPL          GROUP IDGRP4
DESC  JOB2
OVERLIB          STAT CAL
SCHENV          SYSTEM ID          NJE NODE
SET VAR
CTB STEP AT          NAME          TYPE
DOCMEM IDJOB2          DOCLIB
=====
    
```

- 18 Specify **EOW** in the SCHEDULE TAG field. A new SCHEDULE TAG field opens up.
- 19 Specify **EOM** in the new SCHEDULE TAG field, and then specify **EXERCISES** in the third SCHEDULE TAG field that opens up.

This job is now eligible for scheduling if either the EOW or EOM sets of basic scheduling criteria in the Group entity are satisfied. The EXERCISES criteria must also be satisfied to enable you to complete these instructions.

Your entries in the Basic Scheduling parameters section are displayed as follows:

```

=====
SCHEDULE TAG EOW
SCHEDULE TAG EOM
SCHEDULE TAG EXERCISES
SCHEDULE TAG
RELATIONSHIP (AND/OR) 0
DAYS DCAL
AND/OR
WDAYS WCAL
MONTHS 1- Y 2- Y 3- Y 4- Y 5- Y 6- Y 7- Y 8- Y 9- Y 10- Y 11- Y 12- Y
DATES
CONFCAL SHIFT RETRO N MAXWAIT 05 D-CAT
MINIMUM PDS
DEFINITION ACTIVE FROM UNTIL
=====

```

It is unnecessary to specify an IN condition for this job. Although IDJOB1 is a prerequisite to this job, you already defined IDJOB1-ENDED-OK as an IN condition for the entire group.

Therefore, the Runtime Scheduling parameter section should be empty, as shown in [Figure 40](#).

Figure 40 Runtime Scheduling Parameter Section

```

=====
IN
CONTROL
RESOURCE
PIPE
FROM TIME + DAYS UNTIL TIME + DAYS
=====

```

The Post-processing section requires an OUT condition for this job because it is a prerequisite to IDJOB3. But you must decide whether to define Post-processing parameters in case the job ends NOTOK, that is, you need to decide whether the Group entity post-processing parameters are sufficient, or whether IDJOB2 requires its own additional post-processing definitions. For purposes of this exercise, assume that the Group entity Post-processing actions are sufficient.


```

=====
SCHEDULE TAG EOW
SCHEDULE TAG EOM
SCHEDULE TAG EXERCISES
SCHEDULE TAG
RELATIONSHIP (AND/OR) O
DAYS
DCAL
AND/OR
WDAYS
WCAL
MONTHS 1- Y 2- Y 3- Y 4- Y 5- Y 6- Y 7- Y 8- Y 9- Y 10- Y 11- Y 12- Y
DATES
CONFCAL SHIFT RETRO N MAXWAIT 05 D-CAT
MINIMUM PDS
DEFINITION ACTIVE FROM UNTIL
=====

```

Recall that you want this job to be scheduled at end of week and end of month (just like IDJOB2), but only if those days do not fall on a Saturday or Sunday. Therefore, you will keep these Basic Scheduling parameter values, and specify several additional values as well.

Previously, you learned that the SCHEDULE TAG field is followed by a RELATIONSHIP field and a set of Basic Scheduling Criteria:

- The Basic Scheduling criteria allow you to define scheduling criteria apart from the criteria found in the Schedule Tags.
- The RELATIONSHIP field determines the relationship between these criteria and the Schedule Tags. Valid values are A (And) or O (Or):
 - The A value is more restrictive. In addition to the Schedule Tag criteria being satisfied, the basic scheduling criteria of the job, itself, must also be satisfied.
 - The O value is less restrictive. Even if the Schedule Tag criteria of the job are not satisfied, the job can be scheduled if its basic scheduling criteria are satisfied.

You should schedule job IDJOB3 only if the EOM, EOW (or EXERCISES) criteria are satisfied and the day is weekday (Monday through Friday).

25 Specify A (And) in the RELATIONSHIP field.

26 Specify 1,2,3,4,5 in the WDAYs field.

NOTE



You want this job to be scheduled on the day you perform this exercise. Adjust the WDAYs value so that it includes the day on which you are working.

Your entries in the Basic Scheduling parameters section are displayed as follows, except as adjusted to ensure that the WDAY5 field contains your current working day:

```

=====
SCHEDULE TAG EOW
SCHEDULE TAG EOM
SCHEDULE TAG EXERCISES
SCHEDULE TAG
RELATIONSHIP (AND/OR) A
DAYS
DCAL
AND/OR
WDAYS 1, 2, 3, 4, 5
WCAL
MONTHS 1- Y 2- Y 3- Y 4- Y 5- Y 6- Y 7- Y 8- Y 9- Y 10- Y 11- Y 12- Y
DATES
CONF CAL SHIFT RETRO N MAXWAIT 05 D-CAT
MINIMUM PDS
DEFINITION ACTIVE FROM UNTIL
=====

```

There were no Runtime scheduling parameters specified in IDJOB2, so none appear in IDJOB3. But you must add the relevant condition to make this job dependent on IDJOB2.

- 27 Add the IN condition: **IDJOB2-ENDED-OK ODAT**. Your entry in the Runtime Scheduling parameters section is displayed as follows:

```

=====
IN IDJOB2-ENDED-OK ODAT
CONTROL
RESOURCE
PIPE
FROM TIME + DAYS UNTIL TIME + DAYS
=====

```

The Post-processing parameters contain the same OUT condition as IDJOB2. Normally, you would change the string JOB2 to JOB3 in the OUT condition to make the OUT condition correct. But for this exercise you will do something different.

In the exercises you have previously performed in this guide, you defined OUT conditions indicating that the job ended OK, even when that job is not a prerequisite to other jobs. This is good practice, but not a requirement. To demonstrate this, since no other job is dependent on this job, you should delete the condition.

- 28 Delete the OUT condition. The Post-processing parameter section will be empty, and the entire job scheduling definition is displayed as follows:

You can now make the necessary changes.

33 Specify **O** (Or) in the RELATIONSHIP field.

34 Delete the values in the WDAY field.

35 Specify **08,14** and the current working day of the month in the DAYS field.

By defining the current working day in the DAYS field, you can see how the Or Relationship works.

36 Delete the EXERCISES value from the SCHEDULE TAG field.

37 If the day on which you perform these exercises satisfies the EOW or EOM criteria, delete the schedule tags of the satisfied criteria from the job scheduling definition.

The Basic scheduling criteria should only be satisfied by the **DAYS** parameter, not by any schedule tags in the job scheduling definition. If the Or relationship works as expected, the job will still be scheduled.

The IN condition from IDJOB3 now appears in IDJOB4. It should be deleted.

38 Delete the following IN condition: **IDJOB2-ENDED-OK ODAT**.

There was no OUT condition in IDJOB3, so none appears in IDJOB4. You should add the appropriate condition.

39 Add the following OUT condition: **IDJOB4-ENDED-OK ODAT +**. The job scheduling definition for IDJOB4 is displayed as shown in [Figure 41](#).

- 43 Specify **JOB5** in the DESC field. The Basic Scheduling parameters found in IDJOB4 appear in IDJOB5.

You can now make the necessary changes.

- 44 Replace all deleted schedule tags so that the job scheduling definition contains schedule tags EOW, EOM, and EXERCISES.

- 45 Delete the values specified for the DAYS parameter.

The job should only be scheduled according to the specified schedule tags. You should make this job dependent on successful completion of IDJOB4.

- 46 Add the following IN condition: **IDJOB4-ENDED-OK ODAT**.

- 47 In the OUT condition, change the string JOB4 to **JOB5**. The job scheduling definition is displayed as follows:

Notice the following:

- The conditions follow the order in which they were issued:
 - There is an ENDED-OK condition for jobs IDJOB1, IDJOB2, and IDJOB4. (IDJOB3 did not have an OUT condition).
 - There is an END-NOTOK condition for group IDGS2 because of the CC problem in IDJOB5. There is no such condition for IDJOB5 because you did not define it.
 - This is followed by the ENDED-OK for IDJOB5, after the rerun, and finally, the ENDED-OK status is displayed for the group IDGS2.

Advantages of Group Scheduling

Group scheduling is not always desired. But in those situations in which group scheduling is desirable, use of a Group scheduling table simplifies the process and provides several advantages. Some of these are discussed below:

- Schedule tags simplify basic scheduling.

Once a set of basic scheduling parameters is defined in the group entity, all jobs that require that set of scheduling parameters can utilize them simply by specifying the schedule tag name in the appropriate field. The more complex and detailed the scheduling parameters, the greater the advantage.

- Multiple sets of scheduling criteria can be specified for each job.

Regular job scheduling definitions provide for one set of scheduling criteria. However, in a Group scheduling table, the Group entity can include multiple sets of scheduling criteria, and any number of these can be specified in any job scheduling definition in the group. This enables the job to utilize multiple sets of basic scheduling criteria.

- Easier definition of prerequisite conditions and required resources is enabled for each job.

Defining an IN condition in a Group Entity makes all jobs in that group dependent on that condition without defining the IN condition in all the individual job scheduling definitions. The more jobs in the Group table that have no predecessor job other than the predecessor of the Group entity, the greater this advantage.

- Easier Post-processing definition is enabled for each job.

If you want to define Post-processing that should be performed only if any of the jobs ended NOTOK, it is much easier to define the conditional Post-processing once in the Group entity, rather than repeatedly in each job scheduling definition.

If you want to define Post-processing that should be defined if all the jobs ended OK, this is also easier to do in Group tables if there are jobs not belonging to the same hierarchy of dependence.

- Easier triggering of jobs in a different table is facilitated.

If the submission of a job in another table depends on the outcome of all the jobs in the current table, this is easier to define in a Group table.

62 Exit the Online facility.

This completes the steps in this chapter of the *CONTROL-M for z/OS Getting Started Guide*.

Review

In this chapter you

- created a Group scheduling table
- defined a Group entity with multiple sets of scheduling criteria and with a dependency on successful completion of a job from another table
- used Schedule tags to appropriately (according to job) assign sets of scheduling criteria to each job
- defined additional criteria for two of the jobs in the Group table—using an AND relationship with the additional criteria in one of the jobs, and using an OR relationship in the other job
- defined two separate lines of dependency in the jobs in the Group scheduling table—IDJOB3 dependent on IDJOB2, and IDJOB5 dependent on IDJOB4—but no dependency between IDJOB3 and IDJOB4
- ordered the Group entity with the result that all jobs in the table that should have been scheduled that day were also ordered
- observed that the Group entity ended NOTOK when one of the jobs in the Group table ended NOTOK

- learned the advantages of using a Group scheduling table over a regular scheduling table

Recommended Reading

Before continuing with the next chapter, you should read the following in the *CONTROL-M for z/OS User Guide*:

In Chapter 2

- information in the Scheduling Definition facility topics relating to Group tables and Group scheduling
- status changes of the Active Environment screen related to group scheduling and job scheduling definitions in Group tables
- reasons in the Why screen related to Group jobs and entities
- the Group entity in the Zoom screen

In Chapter 3

- summary information related to Group scheduling
- detailed parameter descriptions of the following Group scheduling parameters
 - ADJUST CONDITIONS
 - ON GROUP-END
 - RELATIONSHIP
 - SCHEDULE TAGS
- information that relates to Group scheduling in the following parameters
 - DO statements
 - GROUP
 - MEMNAME
 - ON

AutoEdit and JCL

This chapter includes the following topics:

Overview	134
Preparations	135
Using AutoEdit System Variables in the JCL	135
System Date, Working Date and ODATE	137
Date System Variables	138
Non-Date System Variables	139
Introduction to AutoEdit Control Statements and Functions	140
AutoEdit Function Details	142
%%\$CALCDTE	142
%%\$JULIAN	142
Other Available Functions	144
Supplying Variable Values through the Job Scheduling Definition	144
Checking AutoEdit Syntax and Results	146
Activating Utility CTMAESIM	147
AutoEdit Variables in Other Job Scheduling Definition Statements	150
Supplying Variable Values through an External Member	151
Pointing to External Members	153
Additional AutoEdit Control Statements	154
Complex AutoEdit Terms	155
Operators	156
Local Variables and Global Variables	156
Defining Global Variables	158
Global Variable Syntax	159
Global Variable Access by Another Job	160
Global Variable Update by Another Job	162
JCL Setup Operation Flow	163
Review	165
Recommended Reading	166

Overview

In the production environment, JCL must often be manually modified prior to submission of a job, as in the following cases:

- Changing a parameter or a date card
- Supplying tape numbers in JCL procedures
- Eliminating steps under different run conditions, for example, end of month processing versus normal daily run.

Manual modification of the JCL is inconvenient at best, and it can be error-prone and lead to serious problems. The JCL and AutoEdit facility offers an automated alternative to manual JCL update.

The JCL and AutoEdit facility enables AutoEdit terms, variables, functions, and control statements to be specified in the JCL, in place of values that change from job submission to job submission. AutoEdit terms are prefixed by a pair of percent symbols, %, which distinguishes them from other terms. For example, the term %%ODAY is recognized as an AutoEdit term.

At time of job submission, AutoEdit terms in the JCL are resolved to, or replaced by, their actual values. The inclusion of AutoEdit terms in the job stream can eliminate the need to change JCL once it is defined.

Certain AutoEdit terms can also be used within job scheduling definitions, and can include system and user-defined variables, functions, operators, and control statements.

CONTROL-M provides an AutoEdit simulation facility that enables you to check the results of AutoEdit inclusion before you run a job in the production environment.

You can also define Global AutoEdit variables in a database that allows the variables to be accessed and updated by different jobs.

In this chapter, you will

- see the results of using AutoEdit terms in the JCL of the job, and examine a number of AutoEdit variables
- use one or two AutoEdit functions
- use AutoEdit terms in the job scheduling definition
- look at the online utility facility and use the AutoEdit simulation facility to check the results of our AutoEdit usage

Preparations

For this chapter, do the following:

- Create a new regular scheduling table, not a group table, called IDGS3.
- In table IDGS3, create four very basic job scheduling definitions, for IDJOB6, IDJOB7, IDJOB8 and IDJOB9, as follows:
 - In the MEMNAME field, specify, respectively: **IDJOB6**, **IDJOB7**, **IDJOB8**, or **IDJOB9**.
 - In the GROUP field, specify: **IDGRP5**.
 - In the DESCRIPTION field, specify: **THIS IS IDJOB6** (or, respectively, **IDJOB7**, **IDJOB8**, or **IDJOB9**).
 - In the DAYS field, specify: **ALL**.
 - In the MONTHS field, specify: **Y**.
 - Do not specify any IN or OUT conditions, nor any Runtime Scheduling parameters.
 - Do not specify any Post-processing parameters.
- In your JCL table, create separate JCLs that do nothing—for example, IEFBR14—for jobs IDJOB6, and IDJOB7, IDJOB8, and IDJOB9. Use the JCL from IDJOB1 as the basis for this JCL.

You will make changes to the JCL and job scheduling definitions, as required, during the performance of these exercises.

Using AutoEdit System Variables in the JCL

Suppose that before a job submission you had to update the JCL with the current date, or the jobid of a previously run job. As mentioned in the introduction, such updates are inconvenient at best, and can be a source of errors, as well.

You can use AutoEdit terms to eliminate the need for such updates. In your exercise, you will use several AutoEdit variables in trivial comment lines, but these comment lines will nonetheless demonstrate the usefulness of AutoEdit terms.

- 1 Edit the JCL of IDJOB6, add the following comment lines, and then exit:

```
/** Today's working date is: %%ODATE
/** Today's system date is: %%DATE
/** Today's working day is: %%ODAY
/** The current job order id is: %%ORDERID
```

- 2 Enter the Scheduling Definition facility and order IDJOB6 (in table IDGS3). The Message screen is displayed indicating that the job ran.
- 3 Go to the Active Environment screen and request the Sysout display for the job (enter option V for the job to display the Job Order Execution History screen, and then enter option S to display the Sysout Viewing screen). The Sysout for job IDJOB6 is displayed.
- 4 Page down, or enter the command N (NEXT) in the **COMMAND** field (Command NEXT is discussed in [Chapter 6, “Navigating The Active Environment.”](#)).

The Sysout contains the comment lines you inserted in the JCL, as shown in [Figure 43 on page 136](#)). Notice that the AutoEdit variables in the JCL resolved to their appropriate values.

Figure 43 Sysout for Job IDJOB6

```

----- CONTROL-M/CONTROL-R SYSOUT VIEWING ----- PAGE 2 OF 3
COMMAND ==> SCROLL==> CRSR
MEMNAME IDJOB6 OWNER ID JOBNAME M21 ODATE 020201
-----1-----2-----3-----4-----5-----6-----7-----8
1 //M21 JOB APERLMAN, CLASS=A,
// MSGCLASS=X, NOTIFY=M21,
// MSGLEVEL=(1, 1)
//*----- SUBMITTED BY CONTROL-M (FROM MEMLIB) ODATE=010131
//*----- SCHEDULE CTM. TEST. SCHEDULE(IDGS3)
//*----- JCL CTM. TEST. JCL(IDJOB6)
//*----- CONTROL-M JOB IDENTIFICATION: ORDER ID=001S2 RUN NO.=00001
//* TODAY'S WORKING DATE IS: 010202
//* TODAY'S SYSTEM DATE IS: 010202
//* TODAY'S WORKING DAY IS: 02
//* THE CURRENT JOB ORDER ID IS: 001S2
2 //S1 EXEC PGM=IEFBR14
**** END OF DATA SET - ADDED BY CONTROL M **** DO NOT USE IT IN YOUR PROGRAM !!
ICH70001I STCUSER LAST ACCESS AT 02:00:13 ON FRIDAY, FEBRUARY 02, 2001
ICH70001I STCUSER LAST ACCESS AT 02:00:13 ON FRIDAY, FEBRUARY 02, 2001
IEF142I M21 S1 - STEP WAS EXECUTED - COND CODE 0000
----- - BMC - -----
- CPU (Total) Elapsed CPU (TCB)
-Program Stepname ProcStep RC I/O hh:mm:ss.th hh:mm:ss.th hh:mm:ss.th
COMMANDS: LEFT, RIGHT, FIND str, FIND str PREV, N n, P n, END 03.20.12

```

If you were to run this job again tomorrow, then the variables would resolve accordingly. It should be apparent, that use of AutoEdit system variables such as %%ODATE can eliminate the need to manually update the JCL before a job is run.

Before continuing with the exercise, take a closer look at certain date concepts and AutoEdit date variables.

System Date, Working Date and ODATE

CONTROL-M distinguishes between system date, current working date, and ODATE (original date).

System Date

System date corresponds to the actual calendar date. At any hour on the 9th of September in the year 2000, the system date is the 090900. At the end of the day, when midnight is reached, the system date changes to the 10th of September (100900).

(Of course, your system clock can be set incorrectly. For example, if the system clock is an hour slow, then your system date will not advance at midnight, but rather at 1:00 a.m. Assume that your system clock is set correctly, and that the system time and date correspond to the true time and date.)

Current Working Date

Working date is a CONTROL-M concept that is generally useful in data centers. Basically it allows you to define your day to begin and end at some other time than midnight.

The advantage of using a working day in the production data center is that end of day jobs performed after midnight can have the same working date as jobs performed before midnight; they all have the same working day.

Each site defines its own working day for CONTROL-M. Actually, the site defines the start time of the new working day, which also marks the end of the old working day.

For example, if the new working day at the site begins at 5:00 a.m.:

- A job that runs at 11:00 p.m. on September 9th, 2000 has the following dates (yymmdd format):
 - System date: 000909
 - Current working date: 000909

- A job that runs at 3:00 a.m. on September 9, 2000 has the following dates (yymmdd format):
 - System date: 000910
 - Current working date: 000909

ODATE

You have seen the term ODAT several times when using date references in conditions in IN and OUT and DO COND statements. ODAT is a four-letter abbreviation of ODATE; the term is abbreviated in conditions because the date field of conditions only allows entry of four characters.

ODATE is a keyword that means Original scheduling date, and it refers to (and resolves to) the working date on which the job was originally scheduled.

The important points to note are that:

- ODATE refers to the working date, not the system date.
- The ODATE may in some cases be different than the current working date. For example, if a job was delayed so that it did not run on its ODATE, but instead ran on the next working day, then its current working date would be one day later than its ODATE.

The ODATE allows CONTROL-M (and you) distinguish between different orders of the same job, and determine which job order of the job applies to which date. The same applies to conditions.

Date System Variables

In the preceding steps, you specified a number of variables in the JCL, including, %%ODATE, %%DATE, and %%ODAY. These variables are predefined system date variables, with the following characteristics:

- Predefined system date variables defined to CONTROL-M with recognized meanings.
- Predefined system date variables have values that are not user-supplied, but are resolved from values that are system-supplied.
- Predefined system date variables resolve according to date information.

The AutoEdit facility recognizes many system date variables. These are listed and defined in the JCL and AutoEdit facility chapter of the *CONTROL-M for z/OS User Guide*. However, rather than considering them individually, it is easier to think of system date variables in terms of groupings. Once you have considered the groupings, you can probably anticipate the correct variable name, as illustrated below:

- Date variables can be used for system date, original scheduling date, and current working date.

For example, you saw %%DATE resolve to the system date and %%ODATE resolve to the original scheduling date. You also could have specified %%RDATE, which resolves to the current working date.

While there is no special prefix to indicate system dates, original scheduling date variables have a prefix of %%O, and current working date variables have a prefix of %%R.

- Different date units can be requested.

For example, you saw %%ODAY provide the day unit (1 - 13) of the original scheduling date.

- Some of the more common system date units are %%DAY, %%MONTH, %%YEAR, %%WDAY, which indicates weekday, and %%WEEK, which indicates the week number in the year, 1 through 53.
- The corresponding variables, such as %%OWDAY or %%RWDAY, are available for original scheduling dates and current working dates.
- Other date units of the original scheduling date include %%OMONTH, %%OYEAR, %%OWDAY, for weekday, and %%OWEEK, for week of the year, 1 through 53.

- Julian day variables can also be requested. These are all suffixed JULDAY.

%%JULDAY, %%OJULDAY, and %%RJULDAY are examples of valid Julian date variables.

- Years in the variables described above are 2-character years. If a 4-character year is desired, a \$ should follow the %.

For example, %%\$DATE, %%\$ODATE, and %%\$RDATE each return the appropriated date in yyymmdd format, or a different 4-character year format, depending upon the standards in place at your site.

Non-Date System Variables

In addition to the date system variables discussed above, many system variables that do not relate to dates are available. %%ORDERID, which you specified in the JCL of IDJOB6, and which indicates the ID assigned by CONTROL-M to the job order, is a non-date system variable. Available AutoEdit non-date system variables are listed in the following table:

Table 11 Non-Date AutoEdit System Variables

Variable	Description
%%.	Concatenation symbol.
%%APPL	Application to which the job belongs.
%%BLANK	Blank.
%%BLANKn	Resolves to n blanks, where n is a number between 1 and 99.
%%GROUP	Group to which the job belongs.
%%JOBNAME	Name of the submitted job as specified in the JCL job statement. If %%JOBNAME is resolved before the job submission (for example, %%JOBNAME is used in a SHOUT WHEN LATESUB statement, and the job has not been submitted), its value is assigned the %%\$MEMNAME value.
%%ORDERID	Unique job order ID under CONTROL-M (5 characters).
%%OWNER	Owner of the job, as specified in the scheduling definition.
%%RN	Run number (can exceed one for cyclic and rerun and restarted jobs).
%%TIME	Time of day (format hhmmss).
%%\$MEMNAME	Name of the JCL member from which the job is submitted. (This corresponds to the value specified in the job scheduling definition.)
%%\$QNAME	Qname (unique identifier) of the monitor that submitted the job.
%%\$SCHDLIB	Name of the scheduling library that contains the job scheduling definition of the job.
%%\$SCHDTAB	Name of the scheduling table that contains the job scheduling definition of the job.
%%\$SIGN	1-character ID of the computer on which the job is running.
%%\$TAG	Name of the schedule tag by which the job was scheduled. If the Group scheduling table was forced, or if the job was scheduled based on basic scheduling criteria other than a schedule tag, this value resolves to blanks.

Thus far, you have used AutoEdit predefined system variables. These included date system variables, such as %%ODATE, and non-date system variables, such as %%ORDERID.

Introduction to AutoEdit Control Statements and Functions

- 5 Edit the JCL of IDJOB6, delete the previously added comment lines, and add the following comment lines in their place. Do *NOT* exit the JCL.

```

/** %%SET %%A=%%$CALCDTE %%$ODATE -1
/** THE VALUE OF A IS: %%A

```

You can now examine the components of the preceding %%SET statement.

%% SET

where %%SET is an AutoEdit control statement.

Control statements are used to define the AutoEdit environment, and to control AutoEdit processing, in the JCL. Control statements can appear anywhere in the JCL member to be submitted. When a control statement is detected in a JCL line, for example, in a JCL remark statement, the line containing the control statement is submitted as part of the job. If the control statement appears in a non-JCL line, for example, in a line beginning without a // symbol, the control statement is resolved and the resolved value can be applied to subsequent JCL lines, but the control statement is not submitted as part of the job.

Control statement %%SET sets values of user-defined variables. The format of the statement is:

```
%%SET %%varname=expression
```

where:

- *varname* is the user-defined variable
- *expression* can consist of
 - a value, such as 5
 - a variable, such as %%ODATE
 - a combination of values, variables, operators, functions, and so on, such as %%GENERATION_NUMBER %%PLUS 1

%%A

%%A is a user-defined AutoEdit variable.

Unlike system variables whose names and meanings are predefined and recognized by CONTROL-M, user-defined variables are defined by the user. You could just as easily have called this variable %%FRED or, if you wanted a more meaningful name, %%BACKDATE.

%%\$CALCDTE

%%\$CALCDTE is an AutoEdit function. Like system variables, AutoEdit functions are predefined and have set meanings. In the JCL, AutoEdit functions can only be specified in a %%SET control statement.

Function `%%$CALCDTE` performs date manipulation by adding or subtracting a specified number of days from a specified date.

AutoEdit Function Details

`%%$CALCDTE`

In the preceding step, the principal AutoEdit term is `%%$CALCDTE`. It is true that the function needed to be placed in a `%%SET` control statement, and it is also true that the returned result needed to be represented by a user-defined AutoEdit variable. But the purpose of this `%%SET` control statement is to perform the date calculation function. The control statement and user-defined variable are incidental to the function.

The format of function `%%$CALCDTE` is

`%%$CALCDTE` *date* \pm *quantity*

where

- *date* must be, or must resolve to, a date in format `yyyymmdd`.
- *quantity* is a number, or numeric AutoEdit expression, of days to add or subtract from the date

In this step of your exercise, `%%$CALCDTE` subtracts one day from the `ODATE`.

NOTE



A related function, `%%CALCDATE`, makes the same calculation for dates having a 2-character year, that is, dates resolving to format `yymmdd`. This function has a 366 day limit on the number of days that can be added or subtracted. The `%%CALCDATE` function is supported for reasons of backward compatibility—it is recommended that `%%$CALCDTE` be used whenever possible.

`%%$JULIAN`

You can examine one more function before exiting the JCL.

6 Add the following lines below the `%%$CALCDTE` function in the JCL:

```

/** %%SET %%B=%%$JULIAN %%A
/** THE VALUE OF B IS: %%B

```

Function %%\$JULIAN converts a Gregorian date with a 4-character year, to a Julian date with a 4-character year.

The format of function %%\$JULIAN is:

```
%%$JULIAN date
```

where *date* must be, or must resolve to, a date in format *yyyymmdd*.

In your exercise, you know that %%A, which is defined in the preceding %%\$CALCDTE function, resolves to your ODATE -1. Therefore, %%B should resolve to the Julian version of this date.

- 7 Exit the JCL. You can now run the job.
- 8 Return to the Job List screen and order IDJOB6.
- 9 Go to the Active Environment screen, request the Job Order Execution History screen for the job, then request the Sysout Viewing screen for the job execution, and check the Sysout of the job. Page down to the middle section of the Sysout.

```

**** END OF DATA SET - ADDED BY CONTROL M **** DO NOT USE IT IN YOUR PROGRAM !!
1 //M21 JOB APERLMAN, CLASS=A,
// MSGCLASS=X, NOTIFY=M21,
// MSGLEVEL=(1, 1)
/**----- SUBMITTED BY CONTROL-M (FROM MEMLIB) ODATE=010131
/**----- SCHEDULE CTM. TEST. SCHEDULE(1DGS3)
/**----- JCL CTM. TEST. JCL(1DJOB6)
/**----- CONTROL-M JOB IDENTIFICATION: ORDER ID=001S5 RUN NO. =00001
/** %%SET %%A=%%$CALCDTE 20010202 -1
/** THE VALUE OF A IS: 20010201
/** %%SET %%B=%%$JULIAN 20010201
/** THE VALUE OF B IS: 2001032
2 //S1 EXEC PGM=IEFBR14
**** END OF DATA SET - ADDED BY CONTROL M **** DO NOT USE IT IN YOUR PROGRAM !!

```

Observe the following:

- In the first comment statement, ODATE resolves to the current original scheduling date.
- In the second and third comment lines, %%A resolved to the previous day.
- In the fourth comment line, %%B resolved to the Julian value of %%A.

Other Available Functions

Shown in the following table is a list of some of the other AutoEdit functions available, besides %%\$CALCDTE and %%\$JULIAN. They are described in detail in the *CONTROL-M for z/OS User Guide*.

NOTE



All date-related functions with a \$ symbol, that is, those beginning %%\$, require and return 4-character years, in format yyyy.

Table 12 Other AutoEdit Functions

Function	Description
%%\$GREG	Converts a Julian date to a Gregorian date.
%%\$LEAP	Determines whether a specified Gregorian date falls in a leap year.
%%\$WCALC	Performs a shift from the specified date to a working date in the specified calendar, according to indicated instructions.
%%\$WEEK#	Calculates in which of weeks 1 through 53 a specified date falls. The function uses the site-defined start of the week—Sunday or Monday— as the first day of each week, and assumes that January 1st falls in the first week.
%%\$WEEKDAY	Calculates on which day of the week a specified date falls. The resolved value is an integer from 1 through 6, or 0, where 1 corresponds to the first day of the week—Sunday or Monday, depending on the site-standard—and 0 corresponds to the last day of the week, either Saturday or Sunday.
%%\$YEARWK#	Calculates in which of weeks 1 through 53 a specified date falls, and returns the year and the week number according to ISO8601 standards. The function uses the site-defined start of the week—Sunday or Monday— as the first day of each week, and assumes that the first week is the week that includes January 4th.
%%\$SUBSTR	Extracts a substring from a string.

Supplying Variable Values through the Job Scheduling Definition

As mentioned earlier, AutoEdit variables that resolve to desired values can be placed in the JCL. This is useful at sites where changes to the JCL are discouraged or not allowed.

If the variables are system variables, the values, by definition, are automatically supplied by the system. But if the variables are user-defined variables, the user must somehow provide the values for those variables. One method of providing those values is through the job scheduling definition.

For example, suppose the JCL must be provided the ID of a tape that is being mounted and used. A user-defined AutoEdit variable representing the tape ID can be placed in the JCL. Then, before the job is run, the value for that tape ID can be provided in the job scheduling definition.

You can now add a user-defined variable for the tape ID to the JCL, and set its value in the job scheduling definition. To keep this exercise at a basic level, you will again place the variable in a comment line in a trivial JCL.

- 10 Edit the JCL of IDJOB7, add the following comment line, and then exit the JCL:

```
/** TAPE ID PROVIDED: %%TAPEID
```

You can now edit the job scheduling definition for IDJOB7.

- 11 Enter the job scheduling definition for IDJOB7.

Notice that the General Parameters section contains a field called SET VAR.

```
+-----+
MEMNAME  IDJOB7      MEMLIB  CTM. TEST. JCLLIB
OWNER    ID          TASKTYPE JOB  PREVENT-NCT2  DFLT  N
APPL
DESC     THIS IS IDJOB7
OVERLIB
SCHENV
SET VAR
CTB STEP AT          NAME          TYPE
DOCMEM  IDJOB7      DOCLIB
=====
```

The SET VAR statement is used for specifying a value for a user-defined AutoEdit variable. Assume that the ID of the tape is 046453.

- 12 Specify %%TAPEID=046453 in the SET VAR statement. It appears as follows:

```

JOB:  IDJOB7  LIB CTM. TEST. SCHEDULE          TABLE:  IDGS3
COMMAND  ===>                                SCROLL===>  CRSR
-----+-----
MEMNAME  IDJOB7      MEMLIB  CTM. TEST. JCLLIB
OWNER    ID          TASKTYPE JOB    PREVENT-NCT2  DFLT  N
APPL
DESC     THIS IS IDJOB7
OVERLIB
SCHENV
SET VAR  %%TAPEID=046453
SET VAR
CTB STEP AT      NAME          TYPE
DOCMEM  IDJOB7   DOCLIB
=====

```

- 13** Exit the job scheduling definition, and then exit the Job List screen. Save the changes in the Exit Option window of the Job List screen.

Before running the job, take a look at the AutoEdit Simulation facility.

Checking AutoEdit Syntax and Results

When CONTROL-M detects an AutoEdit syntax error in a JCL member during submission, the submission is canceled by CONTROL-M. Therefore, it is good practice to check the syntax of AutoEdit statements while the member is being prepared.

Furthermore, when the syntax is correct, you may want to verify that the AutoEdit statements return the desired results. For example, you may want to check that you specified the correct AutoEdit date variables for a job that performs end-of-year processing.

The AutoEdit Simulation facility, utility CTMAESIM tests AutoEdit syntax and JCL setup. This utility simulates the actions of the CONTROL-M submission mechanism, which performs AutoEdit processing and JCL setup, and produces a printed report of the process.

Utility CTMAESIM can operate in either JCL Library mode or Scheduling Library mode:

- In JCL Library mode, the utility checks the AutoEdit statements in the JCL of the job, but not in the job scheduling definition.

This mode becomes operational when you specify a JCL member in the utility panel; in this case, the table that contains the job scheduling definition is unknown to the utility, and the job scheduling definition therefore remains unchecked.

- In Scheduling Library mode, the utility not only checks the AutoEdit statements in the JCL of the job, it also checks the impact that SET VAR statements in the job scheduling definition have on the JCL.

This mode becomes operational when you specify a job scheduling definition in the utility panel. Because the job scheduling definition points to the JCL member, both the job scheduling definition and the JCL can be, and are, checked.

Activating Utility CTMAESIM

- 14 Display the Online Utilities menu by selection Option 6 of the Primary Option Menu, under TSO/ISPF.



NOTE

If Option 6 from the IOA Primary Option Menu does not bring you to the Utilities menu, ask your INCONTROL administrator to assist you in accessing the Utilities menu from TSO, or advise you how to access utility CTMAESIM directly.

The IOA Online Utilities menu is displayed. The particular options displayed in the menu depend on the INCONTROL products available at your site, as well as any limitations your INCONTROL administrator may have placed on the display. The IOA Online Utilities menu, and available CONTROL-M utilities, are described in detail in the *INCONTROL for z/OS Utilities Guide*.

```

----- ON-LINE UTILITIES -----
OPTI ON  ===>

                                USERID  -  M21
                                TIME     -  18:06
                                TERMINAL -  3278

I1  PREREQ CONDI TION  - Add/Check/Del ete a Prerequi si te Condi ti on
M1  JOB ORDER ISSUE   - Issue a Job Order
M2  AUTOEDIT SIMUL    - Perform an AutoEdi t Simul ation
M3  SIMUL/TAPE PULL   - Prepare Simul ation/Tape Pul l Li st Job
M4  PARAM PROMPTING   - Parameter Prompting Facili ties
M5  QUI CK SCHEDULE   - Qui ck Schedul e Defini ti on
M6  USER INTERFA CE  - End-User Job Order Interface
R1  CONTROL-R SIMUL   - CONTROL-R Simul ation
R2  DATASET CLEANUP   - CONTROL-R Dataset Cleanu p
R3  JOB DATASET LI ST - Prepare a Job Dataset Li st
R4  STANDALONE        - CONTROL-R Standal one
U1  DOCU/TEXT         - Invoke DOCU/TEXT

X   EXIT              - Exi t Thi s Menu

```

- 15 Enter **M2** in the OPTION field to access utility CTMAESIM. The AutoEdit Simulation panel is displayed.

Figure 44 AutoEdit Simulation Panel

```

----- PERFORM CONTROL-M AUTOEDIT SIMULATION -----
COMMAND ===>

SPECIFY JCL LIBRARY OR SCHEDULE LIBRARY INFORMATION

JCL LIBRARY MODE:
JCL LIBRARY          ===>
MEMBER NAME         ===>
OWNER               ===>
APPLICATION NAME    ===>
GROUP NAME          ===>
SCHEDULE TAG NAME   ===>

SCHEDULING LIBRARY MODE:
SCHEDULING LIBRARY  ===> CTM.TEST.SCHEDULE
TABLE NAME          ===>
JOB NAME            ===>

PARAMETER LIBRARY   ===> CTM.TEST.PARM
WDATE               ===> 02 02 01      (DD MM YY)
ODATE               ===> 02 02 01      (DD MM YY)
FUNCTION            ===> LIST          (LIST/SUBSCAN/SUBMIT)

ENTER YES TO CONTINUE  ===>
    
```

16 Fill in the SCHEDULING LIBRARY MODE parameters with the name of the scheduling library, scheduling table, and job scheduling definition, and enter **YES** in the ENTER YES TO CONTINUE field. Do not change the default values in the GLOBAL LIBRARY, WDATE and ODATE fields; and type **LIST** in the FUNCTION field.

The screen should look similar to the following before you press **Enter**:

```
----- PERFORM CONTROL-M AUTOEDIT SIMULATION -----  
COMMAND ==>  
  
SPECIFY JCL LIBRARY OR SCHEDULE LIBRARY INFORMATION  
  
JCL LIBRARY MODE:  
  JCL LIBRARY           ==>  
  MEMBER NAME          ==>  
  OWNER                 ==>  
  APPLICATION NAME     ==>  
  GROUP NAME           ==>  
  SCHEDULE TAG NAME    ==>  
  
SCHEDULING LIBRARY MODE:  
  SCHEDULING LIBRARY   ==> CTM. TEST. SCHEDULE  
  TABLE NAME         ==> IDGS3  
  JOB NAME             ==> IDJOB7  
  
PARAMETER LIBRARY     ==> CTM. TEST. PARM  
  WDATE                ==> 02 02 01      (DD MM YY)  
  ODATE                ==> 02 02 01      (DD MM YY)  
  FUNCTION              ==> LIST         (LIST/SUBSCAN/SUBMIT)  
  
ENTER YES TO CONTINUE ==> YES
```

The simulation is performed. The output of the simulation is displayed as follows. This is a JCL and AutoEdit scan.

```

***** Top of Data *****
SUB131I SUBMITTED STARTED
04.52.06 AES175I JOB SUBMISSION SIMULATION STARTED
SCHEDLIB CTM.TEST.SCHEDULE
TABLE IDGS3
JOB IDJOB7
WDATE 020201
ODATE 020201
LIST
04.52.06 AESL47I 'SCHEDLIB' CONTROL CARD FOUND - SCHEDULE LIBRARY PROCESSING MOD

04.52.06 AES190I IDJOB7 RETRIEVED FROM TABLE IDGS3 IN SCHEDULE LIBRARY CTM.
04.52.06 AESL53I IDJOB7 RETRIEVED FROM MEMLIB LIBRARY CTM.TEST.JCL

04.52.06 AES177I START SUBMISSION SIMULATION OF MEMBER IDJOB7 LIBRARY CTM.TE
-----
//M21 JOB APERLMAN, CLASS=A,
// MSGCLASS=X, NOTIFY=M21,
// MSGLEVEL=(1,1)
//*NETID=AESUSER
//*---- SUBMITTED BY CONTROL-M (FROM MEMLIB) ODATE=010202
//*---- SCHEDULE CTM.TEST.SCHEDULE(IDGS3)
//*---- JCL CTM.TEST.JCL(IDJOB7)
//*---- CONTROL-M JOB IDENTIFICATION: ORDERID=ZZZZ RUN NO.=00001
//* TAPE ID PROVIDED 046453
//S1 EXEC PGM=IEFBR14
SUB133I IDGRP3 IDJOB7 ZZZZID SUBMITTED FROM LIBRARY (P)
-----
04.52.07 AES178I END SUBMISSION SIMULATION OF MEMBER IDJOB7 LIBRARY CTM.TEST
04.52.07 AES176I JOB SUBMISSION SIMULATION ENDED
SUB159I SHUT DOWN UPON REQUEST OF MAIN TASK
***** Bottom of Data *****

```

Notice that in the comment line you added to the JCL, the user-defined AutoEdit variable is resolved to, or replaced by, the value provided by the SET VAR statement in the job scheduling definition:

//* TAPE ID PROVIDED: 046453

You can now see that if you run this job, the user-defined variable will resolve correctly.

AutoEdit Variables in Other Job Scheduling Definition Statements

In addition to specifying a value for a user-defined AutoEdit variable in a SET VAR statement in the job scheduling definition, a value can also be provided in a DO SET statement. However, because a DO SET statement, like any DO statement, is a conditional Post-processing parameter, it is only activated on a subsequent run of the job, such as a rerun or restart, and then, only provided that the particular ON criteria are satisfied.

AutoEdit terms can also be specified in the following statements in the job scheduling definition:

- **SYSOUT and DO SYSOUT**

File names for SYSOUT and DO SYSOUT handling can be specified using AutoEdit variables whenever SYSOUT Option F, the copy to file or sysout archiving option, is specified, as shown in this example

```
SYSOUT OP F PRM GPL. %%JOBNAME. D%%ODATE. %%JOBID. T%%TIME
```

- **SHOUT, DO SHOUT, and DO MAIL**

System AutoEdit variables can be used in shouted messages, such as

```
MSG JCL ERROR IN JOB %%JOBID %%STEP
```

- **MEMLIB and OVERLIB**

AutoEdit variables can be used in the MEMLIB and OVERLIB fields to specify the appropriate library, as shown in this example

```
OVERLIB CTM. LIB. JCL. OVER%%ODATE
```

Supplying Variable Values through an External Member

In the previous steps, you provided the value for a user-defined AutoEdit variable by updating the job scheduling definition. But you can alternatively provide such values through special members dedicated to this purpose. This enables you to provide the values without updating the JCL, and without updating the job scheduling definition.

You can provide these values through an AutoEdit member identified by the AutoEdit control statements %%LIBSYM and %%MEMSYM.

17 Create a member IDTAPE in a work library. For purposes of this exercise, you will call the library CTM.TEST.AUTOEDIT.

18 In member IDTAPE, place the following lines:

```
%%BRANCH01_TAPE=5554444
%%BRANCH02_TAPE=3336666
%%BRANCH03_TAPE=7654321
```

19 Edit the JCL for IDJOB6 as shown in the following instructions:

- A** Delete the comment lines, that is, those containing AutoEdit terms, that you added earlier.
- B** Add the following statements to the JCL, then exit the JCL:

```

/** %%LIBSYM CTM. TEST. AUTOEDIT %%MEMSYM IDTAPE
/** TAPE ID PROVIDED: %%BRANCHO1_TAPE
/** TAPE ID PROVIDED: %%BRANCHO3_TAPE
    
```

20 Enter the AutoEdit Simulation utility and run the simulation for IDJOB6. The results of the simulation look similar to those shown in [Figure 45](#). Observe the resolved values for the two `/** TAPE ID PROVIDED:` statements.

Figure 45 AutoEdit Simulation Output

```

***** Top of Data *****
SUB131I SUBMITTER STARTED
14.50.17 AES175I JOB SUBMISSION SIMULATION STARTED
SCHEDLIB CTM. TEST. SCHEDULE
TABLE IDGS3
JOB IDJOB6
WDATE 020201
ODATE 020201
LIST
14.50.17 AESL47I 'SCHEDLIB' CONTROL CARD FOUND - SCHEDULE LIBRARY PROCESSING MO
14.50.17 AES190I IDJOB6 RETRIEVED FROM TABLE IDGS3 IN SCHEDULE LIBRARY CTM
14.50.17 AESL53I IDJOB6 RETRIEVED FROM MEMLIB LIBRARY CTM. TEST. JCL
14.50.17 AES177I START SUBMISSION SIMULATION OF MEMBER IDJOB6 LIBRARY CTM. T
-----
//M21 JOB APERLMAN, CLASS=A,
// MSGCLASS=X, NOTIFY=ID,
// MSGLEVEL=(1, 1)
/**NET ID=AESUSER
/**----- SUBMITTED BY CONTROL-M (FROM MEMLIB) ODATE=010202
/**----- SCHEDULE CTM. TEST. SCHEDULE(IDGS3)
/**----- JCL CTM. TEST. JCL(IDJOB6)
/**----- CONTROL-M JOB IDENTIFICATION: ORDER ID=ZZZZ RUN NO.=00001
/** %%LIBSYM CTM. TEST. AUTOEDIT %%MEMSYM IDTAPE
/** TAPE ID PROVIDED: 5554444
/** TAPE ID PROVIDED: 7654321
//S1 EXEC PGM=IEFBR14
SUB133I M21GRP3 IDJOB6 ZZZZM21 SUBMITTED FROM LIBRARY (P)
-----
14.50.18 AES178I END SUBMISSION SIMULATION OF MEMBER IDJOB6 LIBRARY CTM. TEST
14.50.18 AES176I JOB SUBMISSION SIMULATION ENDED
SUB159I SHUT DOWN UPON REQUEST OF MAIN TASK
***** Bottom of Data *****
    
```

Pointing to External Members

In the previous steps, values for user-defined variables were provided in an external member pointed to by a %%LIBSYM / %%MEMSYM control statement. Actually, there are several ways to point to external members, as described in the following table:

Table 13 Pointing to External Members

AutoEdit Term	Description
%%LIBSYM %%MEMSYM	<p>Includes the contents of the specified member at the location of the %%LIBSYM / %%MEMSYM control statement in the JCL. The member is read by CONTROL-M before submission.</p> <p>The included member can only contain a list of AutoEdit variables and their values in the format:</p> <pre>%%term=value</pre> <p>This control statement is useful for providing lists of values to be used during job processing, when both the JCL and the job scheduling definition should not be updated.</p>
%%INCLIB %%INCMEM	<p>Includes the contents of the specified member at the location of the %%INCLIB / %%INCMEM statement in the JCL. The member is read by CONTROL-M before submission. This statement is useful for inserting the following types of information into the JCL:</p> <ul style="list-style-type: none"> ■ JCL statements and/or parameters to be passed to the JCL (for example, SYSIN). ■ AutoEdit control statements, including other %%INCLIB and %%INCMEM statements. <p>Example of usage: If a long segment of JCL must be repeated at different points in the JCL, the segment can alternatively be placed in its own member, and an %%INCLIB / %%INCMEM control statement that points to the member can be specified at the required insertion points.</p>
%%GLOBAL	<p>Includes the contents of the specified member at the location of the %%GLOBAL control statement in the JCL.</p> <p>Like %%MEMSYM, the included member can only contain a list of AutoEdit variables and their values in the format:</p> <pre>%%term=value</pre> <p>Unlike the %%LIBSYM / %%MEMSYM control statement, %%GLOBAL identifies only the member name; it does not identify the library. The specified member is always taken from the library pointed to by DD statement DAGLOBAL.</p>

Additional AutoEdit Control Statements

In addition to the %%SET statement that you used at the beginning of this chapter, and control statements that point to external members, several other AutoEdit control statements are available. These are outlined in the following table:

Table 14 Additional AutoEdit Control Statements

AutoEdit Term	Description
%%LABEL lblnam	%%LABEL creates a tag of name lblnam at the location of the statement.
%%GO lblnam	%%GO lblnam can then be used to pass control to the location identified by the %%LABEL lblnam statement.
%%IF %%THEN %%ELSE %%ENDIF	<p>Provide the AutoEdit facility with Boolean “IF” logic capability. These statements, in conjunction with control statements %%GOTO and %%LABEL, permit branching based on submission time criteria. Job steps, DD statements, and so on are easily excluded or included.</p> <p>Format of the statements is:</p> <pre>%%I F condi ti onal -expressi on statements [%%ELSE] statements %%ENDI F</pre>
%%RANGE	<p>Limits the handling of AutoEdit functions and variables to a specified column range. Contents of all columns outside the range remain unchanged.</p> <p>Format of the statement is:</p> <pre>%%RANGE fromcol tocol</pre>
%%RESOLVE	By default, CONTROL-M must resolve all AutoEdit terms in the JCL or the job is not submitted. This default can be overridden by specifying an appropriate %%RESOLVE statement in the JCL. For details, see the description of the %%RESOLVE statement in the <i>CONTROL-M for z/OS User Guide</i> .

Complex AutoEdit Terms

Multiple AutoEdit variables, and constants, can be joined together into a complex term.

The basic rule of AutoEdit resolution is that when a term contains multiple variables, those variables are resolved from right to left.

Multiple variables can be joined together in the following ways:

- Two variables can be joined to form a single complex variable by linking them together, as follows:

```
%%BACKUP_TAPE%%ODAY%
```

On the third day of the month, the variable partially resolves to %%BACKUP_TAPE03.

If the value of %%BACKUP_TAPE03 is known to CONTROL-M as EE1022, this statement would fully resolve to EE1022

- Two variables can be concatenated into two distinct but joined variables by placing a period between them.

```
%%ODAY. %%OMONTH
```

On the 4th of December, %%ODAY. %%OMONTH resolves to 0412

- Two variables can be concatenated into two distinct variables joined by a period, by placing two periods between them.

```
%%ODAY. . %%OMONTH
```

On the 4th of December, %%ODAY. . %%OMONTH resolves to 04. 12

- A constant can be appended to a variable by prefixing the constant with the concatenation symbol %%.

```
CTM%%MODE%%. 01. JCL
```

If %%MODE=TEST, then CTM%%MODE%%. 01. JCL resolves to: CTM. TEST01. JCL

Operators

AutoEdit operators are used to add or subtract values from AutoEdit variables in the JCL. These operators can only be specified in a %%SET statement. Valid AutoEdit operators are shown in the following table:

Table 15 Valid AutoEdit Operators

Operator	Description
%%PLUS	Adds a value to an AutoEdit variable.
%%MINUS	Subtracts a value from an AutoEdit variable.

AutoEdit operators are generally used as follows:

```
%% SET variabl e=operand operator operand
```

where

- *operand* is an expression that resolves to a numeric value.
- *operator* is %%PLUS or %%MINUS.

Only one operator can be specified in each %%SET statement. Increase the number of generations (%%GENERATION_NUMBER) by one, as follows:

```
// %%SET %%GENERATION_NUMBER=%%GENERATION_NUMBER %%PLUS 1
```

If the value of %%GENERATION_NUMBER was initially 1, it is now set to 2.

Local Variables and Global Variables

User-defined AutoEdit variables are classified as either Local variables or Global variables.

Thus far, all the user-defined variables that you have used in this chapter have been Local variables. A characteristic of Local variables is that values given to them by a job do not carry beyond that job—those values cannot be accessed or changed by a different job.

By contrast, Global variables are stored in the IOA Global Variable database and can be accessed and updated by other jobs.

Shortly, you will create Global variables and demonstrate their global nature. Before doing that, however, you should observe the local nature of the Local variables.

- 21** Enter the job scheduling definition for IDJOB6 and add the following SET VAR statements, and then exit the job scheduling definition:

```
SET VAR    %%LOC1=111
SET VAR    %%BRANCH01_TAPE=222222
```

- 22** Enter the job scheduling definition for IDJOB7 and delete the SET VAR definition. The job scheduling definition now contains no SET VAR definition.

- 23** Edit the JCL for job IDJOB7, and do the following:

- A** Delete the comment line:

```
//* TAPE I D PROVI DED: %%TAPEI D
```

- B** Add, in its place, the following comment lines:

```
//* %%LI BSYM CTM. TEST. AUTOEDI T %%MEMSYM IDTAPE
//* BRANCH01_TAPE HAS THE VALUE: %%BRANCH01_TAPE
//* LET' S SEE I F THI S RESOLVES: %%LOC1
```

- C** Exit the JCL.

- 24** Order the job scheduling definition of IDJOB6.

- 25** After job IDJOB6 has successfully ended, order the job scheduling definition of job IDJOB7.

- 26** Check the results of the job orders in the Active Environment screen. Job IDJOB6 ended "OK", but job IDJOB7 was not submitted.

- 27** Request the log for IDJOB7, by using Option L.

The log indicates that IDJOB7 was not submitted because variable %%LOC1 could not be resolved. Since IDJOB6, which successfully executed, defined a value for %%LOC1, it is clear that IDJOB7 has no access to this value. This is because %%LOC1 is a local variable.

- 28** Edit the JCL for job IDJOB7, delete the following comment line, and then exit the JCL:

```
//* LET' S SEE I F THI S RESOLVES: %%LOC1
```

- 29 Rerun job IDJOB7 through Option R in the Active Environment screen. The second run of IDJOB7 ended “OK”.

The job order line appears as follows in the Active Environment screen:

IDJOB7	ID	020201	M21	/08316	JOB Ended "OK" (Run 2)
					Prior Run: Not Submitted

- 30 Request SYSOUT of job IDJOB7, using Option V in the Active Environment screen followed by Option S in the Job Order Execution History screen, and scroll down to find the value for BRANCH01_TAPE.

%%BRANCH01_TAPE resolved to the original value from %%MEMSYM member, 5554444, not the value set by the SET VAR statement in IDJOB6:

```
//* BRANCH01_TAPE HAS THE VALUE: 5554444
```

Clearly, the SET VAR statement in IDJOB6 did not impact the value in the external member, which indicates that %%BRANCH01_TAPE is a local variable.

Now that you’ve seen the limitations of Local variables, take a look at Global Variables.

Defining Global Variables

As mentioned above, a Global variable is a user-defined variable that is placed in the IOA Global Variable database.

%%SET statements in the JCL, and SET VAR or DO SET statements in the job scheduling definition, enable CONTROL-M jobs and Group entities to define Global variables and place them in the IOA Global Variable database. These variables can then be used and accessed by other jobs. These jobs can use %%SET, SET VAR and/or DO SET statements to change or update the variable values in the database.

You can now define some Global variables in the job scheduling definition and JCL of job IDJOB8.

- 31 Enter the job scheduling definition for IDJOB8 and add the following SET VAR statements. When you define a SET VAR statement and press **Enter**, a new, blank SET VAR statement is opened to allow definition of the next SET VAR statement.

```
SET VAR %%\VAR1=AAA
SET VAR %%. \VAR2=BBB
SET VAR %%. \IDJOB9\VAR1=CCC
```

32 Exit the job scheduling definition.

33 Add the following statements to the JCL of job IDJOB8, and then exit the JCL:

```
//* VAR1 FROM JOB JOB8 EQUALS %%\VAR1
//* VAR2 FROM GROUP GRP3 EQUALS %%. . \VAR2
//* VAR1 FROM JOB JOB9 EQUALS %%. . \IDJOB9\VAR1
```

34 Order the job scheduling definition of IDJOB8.

35 Check the sysout of IDJOB8. The following comments appear in the sysout:

```
//* VAR1 FROM JOB JOB8 EQUALS AAA
//* VAR2 FROM GROUP GRP3 EQUALS BBB
//* VAR1 FROM JOB JOB9 EQUALS CCC
```

From this sysout, you can see that the AutoEdit variables resolved as they should. However, because they appeared in the job scheduling definition and the JCL of the same job, you have not yet demonstrated their global nature.

You will demonstrate the global nature of these variables shortly, but first, take a look at Global variable syntax.

Global Variable Syntax

Note the following points about Global variable assignment and syntax:

- Global variables are identified, and distinguished from Local variables, by a backslash immediately following the %% sign. For example, whereas %%VAR1 is a Local variable, %%\VAR1 is a Global variable.
- A Global variable is assigned an owner at time of creation. This owner can be the component that creates the variable, such as the job, or it can be any other component in the database, such as the job, group, application, or even CONTROL-M. The IOA Global Variable Database has a hierarchical structure to reflect this component hierarchy.
- Backslashes are used to describe the hierarchical structure of the IOA Global Variable Database, much like they are used to describe the directory and subdirectory structure in Unix and DOS.

The full path of the IOA Global Variable database is indicated as follows:

```
%%\product\appl i cati on\group\j ob\vari abl ename
```

- Paths can be specified using the same rules and shortcuts that are available with directories and subdirectories, instead of the full path:
 - A job or Group Entity can assign a Global variable to itself by specifying a backslash immediately following the %% symbol.
 - Paired dots followed by a backslash (..\) indicate movement to the next level up.

Based on the above, you can see that the variables created in IDJOB8 SET VAR statements have the following owners:

Table 16 IDJOB8 SET VAR Statement Owners

Variable	Owner
%%\VAR1=AAA	Global variable %%\VAR1 is owned by job IDJOB8
%%..\VAR2=BBB	Global variable %%\VAR2 is owned by group IDGRP3. (The paired dots moved the variable up the hierarchy to the group to which IDJOB8 belongs.)
%%..\IDJOB9\ VAR1=CCC	Global variable %%\VAR1 is owned by job IDJOB9. (The paired dots move the variable up the hierarchy to group IDGRP3. The \IDJOB9 moves the variable down the hierarchy from IDGRP3 to job IDJOB9.)

- Two variables with the same name but different paths are different variables. This is comparable to the fact that two Unix or DOS files with the same name but different paths are different files.

In our example, %%\VAR1=AAA and %%..\IDJOB9\VAR1=CCC result in different variables. As indicated in the above table

- the Global variable %%\VAR1 that equals AAA belongs to IDJOB8
- the Global variable %%\VAR1 that equals CCC belongs to IDJOB9

- If the particular path has no Group and/or no Application, for example, the job does not belong to a group or application, CONTROL-M utilizes the keyword values NO_APPL and NO_GROUP in the path, as needed.

Global Variable Access by Another Job

You can now take a look at the global nature of these variables.

36 Enter the job scheduling definition for IDJOB9 and do the following:

- A** Enter **ALL** in the DAYS field.
- B** Enter **ANYSTEP** in the ON field.
- C** Enter **C0000** in the CODES field.
- D** Add a DO SHOUT statement whose destination is your user ID, and which contains the following message text:

FIRST SHOUT: %%\VAR1

- E** Add a second DO SHOUT statement whose destination is your user ID, and which contains the following message text:

SECOND SHOUT: %%..\VAR2

- F** Add a third DO SHOUT statement whose destination is your userID, and which contains the following message text:

THIRD SHOUT: %%..\IDJOB8\VAR1

- G** Exit the job scheduling definition.

Before running the job, determine the anticipated results.

Table 17 Anticipated Results of Job

Shout	Anticipated Resolution and Explanation
FIRST SHOUT:	<p>CCC</p> <p>The Shout in IDJOB9 specifies Global variable %%\VAR1, which refers to a Global variable %%\VAR1 that is owned by itself (IDJOB9).</p> <p>Job IDJOB8 assigned the value CCC to the Global variable %%\VAR1 that is owned by IDJOB9.</p>
SECOND SHOUT:	<p>BBB</p> <p>The Shout in IDJOB9 specifies Global variable %%..\VAR2, which refers to a Global variable %%\VAR2 that is owned its group (IDGRP3).</p> <p>Job IDJOB8 assigned the value BBB to the Global variable %%\VAR2 that is owned by group IDGRP3.</p>

Table 17 Anticipated Results of Job

Shout	Anticipated Resolution and Explanation
THIRD SHOUT:	<p>AAA</p> <p>The Shout in IDJOB9 specifies Global variable %%.\IDJOB8\VAR1, which refers to a Global variable %%\VAR1 that is owned by IDJOB8.</p> <p>Job IDJOB8 assigned the value AA A to the Global variable %%\VAR1 that is owned by itself (IDJOB8).</p>

37 Order the job scheduling definition for job IDJOB9.

38 Check the Sysout of the job. The Sysout contains the following:

```
FIRST SHOUT: CCC
SECOND SHOUT: BBB
THIRD SHOUT: AAA
```

These values clearly indicate that these variables have been globally accessed. You can demonstrate that these variables can be globally updated.

Global Variable Update by Another Job

39 Enter the job scheduling definition for IDJOB9 and do the following:

A Add the following SET VAR statements:

```
SET VAR  %%\VAR1=XXX
SET VAR  %%. . \VAR2=YYY
SET VAR  %%. . \IDJOB8\VAR1=ZZZ
```

B Delete the DO SHOUT statements

C Delete the ON STEP / CODES values.

D Exit the job scheduling definition.

The job scheduling definition for job IDJOB9 will now update the values for the Global variables previously defined by IDJOB8.

40 Enter the job scheduling definition for job IDJOB8 and delete the SET VAR statements. Then exit the job scheduling definition.

41 Order job IDJOB9.

42 Once job IDJOB9 has ended, order job IDJOB8.

You should anticipate the following results of these successive job runs.

- The SET VAR statements in job IDJOB9 have updated the Global variables with the new values.
- Because the SET VAR statements were removed from job IDJOB8, this job should not have changed the values that were newly set by IDJOB9.
- Therefore, you can expect that the JCL of job IDJOB8 used the new values set by IDJOB9.

43 Check the sysout of job IDJOB8. The following comments should appear in the sysout:

```
//* VAR1 FROM JOB JOB8 EQUALS ZZZ
//* VAR2 FROM GROUP GRP3 EQUALS YYY
//* VAR1 FROM JOB JOB9 EQUALS XXX
```

The above values clearly indicate that Global variables can be globally updated.

JCL Setup Operation Flow

You have seen that values for variables can come from several sources, for example, external members and SET VAR statements. Therefore, it is important to know the order in which these sources are checked, and the process involved in AutoEdit resolution in JCL.

- All JCL setup operations are performed during job submission. At this time, CONTROL-M processes the JCL of the job line by line.
- CONTROL-M scans each line for AutoEdit terms, which are identified by the %% symbol prefix, and, unless otherwise instructed, tries to resolve them. CONTROL-M resolves all AutoEdit terms in one line before it moves to the next line.
- All changes made during JCL processing, such as variable resolution, are retained only until CONTROL-M has finished submission of the job.
- CONTROL-M resolves system variables by taking the values from the system.
- CONTROL-M resolves Global variables by taking the values from the IOA Global Variable database.

Values for Local user-defined variables can be taken from any of several possible sources, as described below. When CONTROL-M detects a Local user-defined variable in the JCL line being processed, it checks these possible sources in a specific order until a value is found for the variable. CONTROL-M creates a user-defined variable environment in which it places each user-defined variable and its value.

The potential sources for Local user-defined variable values are listed below in the order in which they are generally checked:

- system variable values

- %%SET control statements

These statements can be specified in JCL lines, including JCL comment lines. They assign values to variables.

- SET VAR and DO SET statements

These statements can be specified in the job scheduling definition. They can be used to define new variables, or to assign new values to existing variables.

SET VAR statements can affect all job submissions.

DO SET statements can override values specified by a SET VAR or previous DO SET statement. However, since DO SET statements are Postprocessing parameters, they only affect subsequent runs of a job, specifically, rerun and restarted jobs.

- Local variables and values defined in members specified in %%LIBSYM / %%MEMSYM control statements.

These members define local variables and specify their values. Members are searched in the order they appear in the JCL.

- Local variables and values defined in members specified in %%GLOBAL control statements.

These members define local variables and specify their values. Members are searched in the order they appear in the JCL.

The order in which CONTROL-M checks potential sources for possible AutoEdit variable resolution is important because once CONTROL-M has resolved a variable, it generally stops checking other sources. Potential values from other sources are ignored, and resolved values are not overridden except by %%SET statements in subsequent JCL lines.

Because JCL is processed sequentially one line at a time, the line being processed can only be affected by external members and %%SET control statements that have previously been processed. If a line contains an undefined variable that is only defined in a subsequent line, the variable cannot be resolved.

By default, if CONTROL-M cannot resolve a variable, it stops submission of the job. This default, however, can be overridden by specifying the %%RESOLVE control statement with a value of NO or OFF.

44 Exit the Online facility.

This completes the steps in this chapter of the *CONTROL-M for z/OS Getting Started Guide*.

Review

In this chapter you

- looked at and used AutoEdit system variables (date and nondate variables) in the JCL of a job
- learned about the difference between System date, Current Working date, and ODATE
- used the AutoEdit %%SET control statement in your JCL, and learned about other AutoEdit control statements
- used AutoEdit functions %\$CALCDTE and %\$JULIAN in your JCL, and learned about other AutoEdit functions
- used the SET VAR statement in the job scheduling definition to supplied values to user-defined variables in the JCL of your job
- checked AutoEdit syntax and results using online utility CTMAESIM
- supplied values to AutoEdit variables through an external member that was pointed to by the %%LIBSYM and %%MEMSYM statement, and learned other ways to point to external members
- learned about complex AutoEdit terms and AutoEdit resolution
- learned about AutoEdit operators
- learned the difference between Local and Global variables

- learned the syntax of, and defined and updated Global AutoEdit variables
- learned JCL Setup Operation flow

Recommended Reading

Before continuing with the next chapter, you should read the following in the *CONTROL-M for z/OS User Guide*:

- in Chapter 1, the topic dealing with AutoEdit
- in Chapter 2, the description of utility CTMAESIM
- in Chapter 3, description of the SET VAR and DO SET statements
- in Chapter 5, the entire chapter

Navigating The Active Environment

This chapter includes the following topics:

Overview	168
Preparations	168
A Closer Look At Filters	169
Displaying the List of Available Filters	173
Changing Active Environment Display Types	175
Adding a Note to a Job	176
Displaying Table Names for Jobs	179
Displaying Job Dependencies	180
Displaying Execution Information from Job Runs	182
Job Order Execution History Screen	182
Sysout Viewing Screen	184
Forcing an OK Status for a Job	186
Displaying Statistics for a Job	188
Displaying Jobs Belonging to a Specific Group	189
Review	190
Recommended Reading	190

Overview

You are now going to look at and use some additional options, screens and capabilities of the Active Environment.

In this chapter, you will

- look more closely at the fields in the Show Screen Filter window, define a new filter, and display a list of available filters
- change between display types of the Active Environment screen
- add a note to a job in the Zoom screen and display the content of the note in the Active Environment screen
- display the scheduling library and table names of jobs in the Active Environment screen
- display the list of dependent jobs, both predecessors and successors, in the Job Dependency Network screen
- display the execution history of job orders in the Job Order Execution History screen, and display job Sysout information in the Sysout Viewing screen
- force a status of OK for a job in the Active Environment screen
- display statistics for a job in the Statistics screen
- display the Group to which a job belongs in the Active Environment screen

Preparations

In this chapter, you will be using the following, and their respective JCLs:

- IDJOB1 from table IDGS1
- Each job, from IDJOB2 through IDJOB5, and the table IDGS2 Group entity
- IDJOB6 from table IDGS3

In preparation, you will

- ensure that job IDJOB1 in table IDGS1 does not require manual confirmation

- ensure a continuous job dependency from IDJOB1 through IDJOB6, by
 - making the following changes in the appropriate job scheduling definitions in scheduling table IDGS2
 - adding, in the job scheduling definition of IDJOB3, OUT condition | DJOB3-ENDED-OK ODAT +.
 - adding, in the job scheduling definition of IDJOB4, IN condition | DJOB3-ENDED-OK ODAT
 - making the following changes in the appropriate job scheduling definitions in scheduling table IDGS3
 - adding, in the job scheduling definition of IDJOB6, IN condition | DJOB5-ENDED-OK ODAT
 - adding, in the job scheduling definition of IDJOB6, OUT condition | DJOB6-ENDED-OK ODAT +
- ensure the following situations in the appropriate JCL files:
 - in the JCL of job IDJOB3, the step returns a condition code of C0008
 - in the JCL of job IDJOB5, all steps return a condition code of C0000
- ensure that batch utility CTMJSA has been run in the current working day

If CTMJSA is not run as part of New Day processing, run it yourself or request that it be run by your administrator

A Closer Look At Filters

In anticipation of taking a closer look at filters, you can run the jobs you prepared, and then look at the results in the Active Environment screen.

- 1 Order the following jobs, in the following order:
 - Job IDJOB1 in table IDGS1
 - Group Entity GRPSCHD in table IDGS2
 - Job IDJOB6 in table IDGS3
- 2 Enter the Active Environment screen and activate the filter IDGS. The following screen is displayed:

```

Filter: IDGS          ----- CONTROL-M Active Environment ----- UP <D> - (3)
COMMAND ==>                                     SCROLL ==> CRSR
O Name      Owner      Odate  Jobname  JobID   Typ ----- Status -----
IDJOB1     ID           020201 M21      /08915  JOB Ended "OK"
GRPSCHD    ID           020201          GRP Active - In Error
IDJOB2     ID           020201 M21      /08916  JOB Ended "OK"
IDJOB3     ID           020201 M21      /08917  JOB Ended- Not "OK" Due to CC
IDJOB4     ID           020201          JOB Wait Schedule
IDJOB5     ID           020201          JOB Wait Schedule
IDJOB6     ID           020201          JOB Wait Schedule
===== >>>>>>>>>>>> Bottom of Jobs List <<<<<<<<<<<<<< =====

Commands: OPt Display Show HIstory RBal REFresh Auto Jobstat SHPF Note Table
          OPt command toggles between Commands and Options display 14.13.03
    
```

These results are not surprising. Job IDJOB3 did not end OK due to condition code C0008. And because of the defined job dependencies, the remaining jobs have a status of Wait Schedule.

You can now take a closer look at filters in the Active Environment screen. In [Chapter 1, "Introduction to CONTROL-M,"](#) you created filter IDGS, and in subsequent chapters, you activated this filter by entering the SHOW IDGS command.

To display the filter criteria of this same filter for editing, you should add the keyword EDIT at the end of the command. You can now edit filter IDGS.

3 Enter the command **SHOW IDGS EDIT**. The Show Screen Filter window displays the filtering criteria for filter IDGS.

Figure 46 Show Screen Filter Window

```

----- Show Screen Filter -----(3. SHOW)-
Filter IDGS          Save  (Y/N) Desc: GS-EXERCISES
Memname
Group  IDGRP
===== In Process Y ===== | Ended          Y | ===== State Y =====
-----
Wait Sched  Y  Wait time  Y | Ended "OK"   Y | Free        Y  Forced OK Y
Wait Conf  Y  Wait Cond  Y | Not "OK"     Y | Held        Y  Grp Held  Y
Wait SUB   Y  Wait quant Y | Rerun        Y | On Req      Y  CMEM Forc Y
Submitted  Y  Wait contrl Y | Disappeared  Y | Deleted     N  Note      Y
Wait Exec  Y  Grp Active  Y | Abended      Y | Late        Y  Restarted Y
Executing  Y                               | Unexpected CC Y | Pseudo      Y
On Out Queue Y                               | JCL Error    Y |
Task Type: Job Cyc Emr Stc Cst Est Ecj Ecs Wrn Grp
           Y  Y  Y  Y  Y  Y  Y  Y  Y  Y
Res Name
Resource Type: In Y Out Y Conds Y Resource Y Control Y
Owner
Odate:  From                To                Priority
Job
CPU Id          LPAR
Sch Lib
-----
OPt command toggles between Commands and Options display 11.07.49

```

You have already seen that you choose whether to save a new filter by specifying **Y**, or **N**, in the **Save** field, and pressing **Enter**. You can use this same field to choose whether to save changes to an existing filter.

NOTE



Specifying **N** (No) does not cancel changes made to a filter. It only means that they will not be permanently saved. They will, however, remain in memory. This applies even if you are editing a new filter, that is, specifying **N** and exiting the filter leaves the new filter in memory.

To cancel changes to a filter, close the window by pressing **PF04/PF16** to enter the **RESET** command. The changes are canceled regardless of the value specified in the **Save** field.

The purpose in filtering the display was to ensure that the screen only displays those jobs that you used in your exercises. You took two steps to accomplish this purpose

1. You ensured that all jobs in the exercises belonged to a Group prefixed by **IDGRP**.
2. You then filtered the display based on this Group name prefix of **IDGRP**.

You could have accomplished the purpose using the following filter criteria:

- Because you prefixed each member name with **IDGS**, you could have filtered using that **Memname** prefix.
- If you ran no other jobs, you could have filtered on **Owner ID**.

Clearly, filtering on Odate would not have accomplished your purpose. However, Odate is useful for filtering when you only want to see jobs scheduled for a specific Odate.

The middle portion of the window is divided into three columns. These columns are all status related. For example, you can include (or exclude) jobs that have a Wait Schedule, Wait Confirmation, and/or Ended "OK" status. To learn the details of the relationship between these status columns, and their header topics, In Process, Ended, State, refer to the description in the *CONTROL-M for z/OS User Guide*.

If you filter on such values as a common member name or group prefix, you are likely to display related or connected jobs. By contrast, if you filter only on a status such as Wait Schedule, you are likely to see completely unrelated jobs in the display. This, however, is still very valuable. An operator, for example, may need to check which jobs still have a Wait Schedule or a Wait Confirmation status.

Multiple filtering criteria can be specified. All specified criteria must be satisfied. You can, for example, specify a filter on your Owner name and on a Wait Schedule status. This way, you can see only your jobs that are Waiting Scheduling.

The filter window for filter IDGRP is currently displayed. You can, of course, close it without making changes. And, as mentioned above, you can make changes and either save them or keep them in memory.

But you can also use this filter as the basis for another filter by making the desired changes and specifying a new name in the Filter field.

You can define a new filter that shows only jobs with a group name prefix of IDGRP that have a Wait Schedule status. Name this filter IDWS, as described in the following steps.

- 4 In the Filter field, change the name from IDGS to **IDWS**.
- 5 Specify **Y** in the Save column.
- 6 Change the description to read: **GS-WAITSCHED**.
- 7 Leave the Group name value, IDGRP, unchanged.
- 8 In the status sections leave the In Process value (Y) and the Wait Sched value (Y) unchanged. Also, leave all values in the State column unchanged.
- 9 In the status sections, make the following changes:
 - A In the In Process column, change all values except those of In Process and Wait Schedule to **N**.
 - B Change the Ended column header value to **N**.

Figure 48 Display Type A (All Fields)

```

Filter: IDGS          ----- CONTROL-M Active Environment ----- UP <A> - (3)
COMMAND ==>                                               SCROLL ==> CRSR
O Name      Owner      Odate  Jobname  JobID   Typ ----- Status -----
IDJOB1     ID          020201 M21      /26288  JOB Ended "OK"
  OrderID 001M6  Grp IDGRP1
  MaxRC
  Time Fr:          Time Un:          Res. Use: Y
  Due-In: 1158  Due-Out: 1159  Priority:
  Avg Elaps: 0001          RBA: 00013A
GRPSCHD   ID          020201          GRP Active - In Error
  OrderID 001M7  Grp IDGRP4          G
  MaxRC
  Time Fr:          Time Un:          Res. Use: Y
  Due-In: 1156  Due-Out: 1159  Priority:
  Avg Elaps: 0003          RBA: 00013C
IDJOB2     ID          020201 M21      /26289  JOB Ended "OK"
  OrderID 001M8  Grp IDGRP4          G
  MaxRC
  Time Fr:          Time Un:          Res. Use: Y
  Due-In: 1158  Due-Out: 1159  Priority:
  Avg Elaps: 0001          RBA: 00013E
IDJOB3     ID          020201 M21      /26290  JOB Ended- Not "OK" Due to CC
Commands: OPT Display Show History RBAI REFRESH Auto Jobstat SHPF Note Table
          OPT command toggles between Commands and Options display 14.35.40
    
```

Notice that this display type includes many fields not displayed in the Default display type. Because of the larger number of lines displayed for each job, each screen display shows far fewer jobs than the default display type.

15 Enter **DI D** in the COMMAND field. The Default display type is redisplayed.

You have not displayed the Net display type, **DI N**. This display type is generally useful only in the Job Dependency Network screen, which is discussed, below, under the topic “Displaying Job Dependencies.”

Next, you will see how to add a note to one of your jobs.

Adding a Note to a Job

The NOTE command that appears in the list of commands at the bottom of the Active Environment screen is *NOT* used to add notes. Rather, it is used to display notes that have already been added to a job order.

Notes are added to job orders in the Zoom screen. Remember that before making any changes in the Zoom screen, you must first place the job in Held status.

16 Place job IDJOB6 in Held status and display the Zoom screen for the job by entering Option H, and pressing **Enter** until the job has a status of Held Wait Schedule. Then enter option Z).

- 17** Enter command **NOTE** in the **COMMAND** field of the **Zoom** screen. A **Note** line is opened inside the **Zoom** screen between two lines of equal signs, as follows:

```

----- CONTROL-M ZOOM SCREEN ----- (3. Z)
COMMAND ==> SCROLL==> CRSR
-----+-----
MEMNAME I DJOB6      MEMLI B  CTM. TEST. JCL
OWNER   I D          TASKTYPE JOB          PREVENT-NCT2  DFLT- N
SCHDTAB I DGS3      SCHDLI B  CTM. TEST. SCHEDULE
APPL                                GROUP I DGRP5
OVERLI B                                STAT CAL
SCHENV                                NJE NODE
JOBNAME          JOBI D      ODATE O20201 ORDERI D 001MC  MAXWAI T 05
RESTART DECI SI ON-FROM      .      TO      .      CONFIR M N
DESC
SET VAR
CTB STEP AT      NAME          TYPE
DOCMEM I DJOB6      DOCLI B
=====
NOTE
=====
IN          I DJOB5-ENDED-OK      O202
CONTROL
RESOURCE
PIPE
TIME: FROM      UNTIL      PRI ORI TY      CONFIR M N
COMMANDS:  SAVE  DOC  NOTE      20.51.11

```

When you enter text in one line, a new blank line is opened to enable you to enter more text.

- 18** In the **NOTE** line, enter the text: **THIS NOTE COULD BE IMPORTANT. BUT IT'S NOT.**

The **NOTE** section is displayed as follows:

```

=====
NOTE THIS NOTE COULD BE IMPORTANT. BUT IT'S NOT.
NOTE
=====

```

Save this new note by typing **SAVE** in the command line and pressing **Enter**.

- 19** Exit the **Zoom** screen. The **Active Environment** screen is displayed.

A **Note** flag, showing the word *****NOTE**** in the **STATUS** field for **IDJOB6**, indicates that the job has an appended note.

This is because the NOTE command acts as a toggle between displaying and hiding the text of each note appended to a job order.

22 Free job IDJOB6, using Option F.

Displaying Table Names for Jobs

Notice the TABLE command at the bottom of the screen. This command toggles between displaying and hiding the name of the scheduling library and table to which each job order belongs.

23 Enter the TABLE command. The name of the scheduling library and table are displayed for each job in the Status field.

```

Filter: IDGS          ----- CONTROL-M Active Environment ----- UP <D> - (3)
COMMAND ==>>>                                           SCROLL ==> CRSR
O Name      Owner      Odate  Jobname  JobID   Typ ----- Status -----
IDJOB1     ID          020201 M21      /08915  JOB Ended "OK"
                                     SCHED-LIB=CTM. TEST. SCHEDULE
                                     (IDGS1)
GRPSCHD    ID          020201                GRP Active - In Error
                                     SCHED-LIB=CTM. TEST. SCHEDULE
                                     (IDGS2)
IDJOB2     ID          020201 M21      /08916  JOB Ended "OK"
                                     SCHED-LIB=CTM. TEST. SCHEDULE
                                     (IDGS2)
IDJOB3     ID          020201 M21      /08917  JOB Ended- Not "OK" Due to CC
                                     SCHED-LIB=CTM. TEST. SCHEDULE
                                     (IDGS2)
IDJOB4     ID          020201                JOB Wait Schedule
                                     SCHED-LIB=CTM. TEST. SCHEDULE
                                     (IDGS2)
IDJOB5     ID          020201                JOB Wait Schedule
                                     SCHED-LIB=CTM. TEST. SCHEDULE
                                     (IDGS2)
IDJOB6     ID          020201                JOB Wait Schedule
Commands: OPT Display Show History RBal REFresh Auto Jobstat SHPF Note Table
          OPT command toggles between Commands and Options display      21.29.12
    
```

24 Enter the TABLE command again. The names of the scheduling libraries and tables are hidden.

The Job Dependency Network screen is a special case of the Active Environment screen. It indicates predecessor and successor jobs, and the levels of those jobs, relative to the selected job.

Note the following about the jobs listed in the screen:

- An arrow appears beside the job for which the N option was requested. This job is the starting point for looking at predecessor and successor jobs.
- Immediately below IDJOB5 is its the only successor job, IDJOB6, with a level number of +1. The plus sign indicates that IDJOB6 is a successor job; the integer 1 indicates that IDJOB6 is only one level removed from IDJOB5, that is, it is an immediate successor.
- Above IDJOB5 is the list of predecessor jobs, each displaying a negative level number. The minus signs in the level numbers indicate that these are predecessor jobs of IDJOB5. The integer in the level number indicates the number of levels the job is removed from IDJOB5. For example, IDJOB4, level -1, is the immediate predecessor; IDJOB2, level -3, is three levels away from IDJOB5.
- Jobs appear in the screen in job flow order, from earliest predecessor to latest successor.

Job dependencies are determined according the prerequisite IN and OUT conditions of the job. DO COND conditions are ignored because they are conditional rather than constant.

The screen also displays the following information about the jobs, much of it also found in the Zoom screen:

Table 18 Job Dependency Fields (Part 1 of 2)

Field	Description
DueIN	Due in time. Time by which the job must be submitted.
DueOut	Due out time. Time by which the job must finish executing.
Elaps	Elapse time. Expected time (in minutes) for the job to execute.
Late	Indication that a job is late. Possible values: <ul style="list-style-type: none"> ■ X – Actual execution has not completed within the expected execution time. Also indicates that SHOUT WHEN EXECTIME was issued. ■ I – Job was not submitted in time. Also indicates that SHOUT WHEN LATESUB was issued. ■ O – Job is late. Also indicates that SHOUT WHEN LATE was issued.
Prio	CONTROL-M priority of the job.

Table 18 Job Dependency Fields (Part 2 of 2)

Field	Description
Res	Indicator that the job accesses Quantitative resources. Valid values are: <ul style="list-style-type: none"> ■ blank – Quantitative resources are not accessed. ■ Y – Quantitative resources are accessed.
Status	Job (task) status.

To refresh the display with the latest information, enter the REFRESH command as you did before displaying the screen. The time of the last refresh is displayed on the top line of the Job Dependency Network screen.

You can change display types in the Job Dependency Network screen, but there is little point in doing so because it is this display type that shows the dependency levels.

- 28** Press PF03/PF15 to exit from the Job Network Dependency screen back to the Active Environment screen.

Displaying Execution Information from Job Runs

Job Order Execution History Screen

You can now display the execution history for job IDJOB1. As you saw in [Chapter 1, “Introduction to CONTROL-M,”](#) you can do this by specifying Option V (View Sysout) for the job.

- 29** Enter option V for job IDJOB1. The Job Order Execution History screen for IDJOB1 is displayed.



NOTE

This facility requires that CONTROL-M/Restart be operational at your site. If CONTROL-M/Restart is not installed, skip to “Forcing an OK Status for a Job” below.

```

----- JOB ORDER EXECUTION HISTORY -----(3.V)
COMMAND ===>                                SCROLL===> CRSR
MEMNAME IDJOB1      OWNER ID                ORDERID 000EW  ODATE 020201
O JOBNAME  JOBID  DATE  START  ELAPSED  PAGES  MAX RC  -----  STATUS  -----
  M21      08915 020201 19:27   0:00  00003      ENDED "OK"
===== >>>>>>>>> BOTTOM OF ACTIVE JOB ORDER HISTORY LIST <<<<<<<<<< =====
    
```

OPTION: S SELECT

00.00.34

The following information about the job is displayed at the top of the screen:

Table 19 Job Information

Field	Description
MEMNAME	Name of the member containing the JCL of the job.
OWNER	User ID of the owner of the job.
ORDERID	Job order ID.
ODATE	Original scheduling date of the job.

The following information is provided for each execution of the job:

Table 20 Job Execution Information

Field	Description
JOBNAME	Job name.
JOBID	JES job number.
DATE	Execution date of the job.
START	Start time of the job execution, in format hh:mm.
ELAPSED	Total elapsed time of the job execution, in format mmmm.nn, where <i>mmmm</i> is minutes, and <i>nn</i> is hundredths of minutes).
PAGES	Number of pages in the sysout.
MAX RC	Highest return code of the job execution.
STATUS	Status assigned to the job by CONTROL-M, based on execution results.



NOTE

The above display type is the Default display type. Your INCONTROL administrator can create additional display types. To change display types, entering the command **DI x**, as you did to change display types in the Active Environment screen.

Sysout Viewing Screen

Notice the option S (Select) at the bottom of the screen. You can specify this option to display the sysout of the selected job run. Alternatively, you can enter VIEWALL in the COMMAND field to display the sysout of all runs of the job.

30 Enter option **S** for the job run of IDJOB1. The Sysout Viewing screen is displayed.

Figure 49 illustration is obviously more than one screen length. It shows the full sysout.

Figure 49 Sysout Viewing Screen

```

----- CONTROL-M/CONTROL-R SYSOUT VIEWING ----- PAGE 1 OF 3
COMMAND ==> SCROLL==> CRSR
MEMNAME IDJOB1 OWNER ID JOBNAME M21 ODATE 020201
-----1-----2-----3-----4-----5-----6-----7-----8
1 JES2 JOB LOG -- SYSTEM OS35 -- NO
0
18.05.20 JOB25002 ---- FRIDAY, 02 FEB 2001 ----
18.05.20 JOB25002 IRR010I USERID STCUSER IS ASSIGNED TO THIS JOB.
18.05.20 JOB25002 ICH70001I STCUSER LAST ACCESS AT 18:03:22 ON FRIDAY, FEBRUA
18.05.20 JOB25002 ICH70001I STCUSER LAST ACCESS AT 18:03:22 ON FRIDAY, FEBRUA
18.05.20 JOB25002 $HASP373 M21 STARTED - INIT 3 - CLASS A - SYS OS35
18.05.20 JOB25002 IEF403I M21 - STARTED - TIME=18.05.20
18.05.20 JOB25002 - CPU (Total) Elapse
18.05.20 JOB25002 -Jobname Stepname ProcStep RC I/O hh:mm:ss.th hh:mm:
18.05.20 JOB25002 -M21 S1 00 0 00.02
18.05.20 JOB25002 IEF404I M21 - ENDED - TIME=18.05.20
18.05.20 JOB25002 -
18.05.20 JOB25002 -M21 Job Service Totals 0 00.02
18.05.21 JOB25002 $HASP395 M21 ENDED
0----- JES2 JOB STATISTICS -----
- 02 FEB 2001 JOB EXECUTION DATE
- 8 CARDS READ
- 44 SYSOUT PRINT RECORDS
- 0 SYSOUT PUNCH RECORDS
- 3 SYSOUT SPOOL KBYTES
- 0.00 MINUTES EXECUTION TIME
**** END OF DATA SET - ADDED BY CONTROL M **** DO NOT USE IT IN YOUR PROGRAM !!
1 //M21 JOB APERLMAN, CLASS=A,
// MSGCLASS=X, NOTIFY=ID,
// MSGLEVEL=(1,1)
//*----- SUBMITTED BY CONTROL-M (FROM MEMLIB) ODATE=010202
//*----- SCHEDULE CTM.TEST.SCHEDULE(IDGS1)
//*----- JCL CTM.TEST.JCL(IDJOB1)
//*----- CONTROL-M JOB IDENTIFICATION: ORDER ID=001JW RUN NO.=00001
2 //S1 EXEC PGM=IEFBR14
**** END OF DATA SET - ADDED BY CONTROL M **** DO NOT USE IT IN YOUR PROGRAM !!
    
```



```

Filter: M21GS      ----- CONTROL-M Active Environment ----- UP <D> - (3)
COMMAND ==>>>
O Name      Owner      Odate  Jobname  JobID   Typ ----- Status -----
  IDJOB1    ID          020201 M21      /26288  JOB Ended "OK"
  GRPSCHD   ID          020201          GRP Active - In Error
  IDJOB2    ID          020201 M21      /26467  JOB Ended "OK"
O IDJOB3    ID          020201 M21      /26468  JOB Ended- Not "OK" Due to CC
  IDJOB4    ID          020201          JOB Wait Schedule
  IDJOB5    ID          020201          JOB Wait Schedule
  IDJOB6    ID          020201          JOB Wait Schedule
                                     *** Note ***
===== >>>>>>>>>>>> Bottom of Jobs List <<<<<<<<<<<< =====

```

35 Free IDJOB3. The status of IDJOB3 changes to **Ended "OK" Forced OK**.

This, in turn, results in the IDJOB3-ENDED-OK condition to be added to the IOA Conditions file, which in turn allows the remaining jobs (IDJOB4, IDJOB5, and IDJOB6), and the Group entity to successively end OK.

In the following screen illustration, the status of IDJOB3 has already changed, and IDJOB4 has been changed to Wait Submission.

```

Filter: IDGS      ----- CONTROL-M Active Environment ----- UP <D> - (3)
COMMAND ==>>>
O Name      Owner      Odate  Jobname  JobID   Typ ----- Status -----
  IDJOB1    ID          210101 M21      /26288  JOB Ended "OK"
  GRPSCHD   ID          210101          GRP Active
  IDJOB2    ID          210101 M21      /26467  JOB Ended "OK"
  IDJOB3    ID          210101 M21      /26468  JOB Ended "OK" Forced OK
  IDJOB4    ID          210101          JOB Wait Submission
  IDJOB5    ID          210101          JOB Wait Schedule
  IDJOB6    ID          210101          JOB Wait Schedule
                                     *** Note ***
===== >>>>>>>>>>>> Bottom of Jobs List <<<<<<<<<<<< =====

```

At the end of the process, the screen is displayed as follows:

```

Filter: IDGS      ----- CONTROL-M Active Environment ----- UP <D> - (3)
COMMAND ==>>>
O Name      Owner      Odate  Jobname  JobID   Typ ----- Status -----
  IDJOB1    ID          020201 M21      /26288  JOB Ended "OK"
  GRPSCHD   ID          020201          GRP Ended "OK"
  IDJOB2    ID          020201 M21      /26694  JOB Ended "OK"
  IDJOB3    ID          020201 M21      /26695  JOB Ended "OK" Forced OK
  IDJOB4    ID          020201 M21      /26696  JOB Ended "OK"
  IDJOB5    ID          020201 M21      /26697  JOB Ended "OK"
  IDJOB6    ID          020201 M21      /26693  JOB Ended "OK"
                                     *** Note ***
===== >>>>>>>>>>>> Bottom of Jobs List <<<<<<<<<<<< =====

```


43 Exit the Online facility.

This completes the steps in this chapter of the *CONTROL-M for z/OS Getting Started Guide*.

Review

In this chapter you

- looked more closely at the fields in the Show Screen Filter window, defined a new filter, and displayed a list of available filters
- changed between display types of the Active Environment screen
- added a note to a job in the Zoom screen and displayed the contents of the note in the Active Environment screen
- displayed the scheduling library and table names of jobs in the Active Environment screen
- displayed the list of dependent jobs (predecessor and successor jobs) in the Job Dependency Network screen
- displayed the execution history of job orders in the Job Order Execution History screen, and display job sysout in the Sysout Viewing screen
- forced a status of OK for a job in the Active Environment screen
- displayed Statistics for a job in the Statistics screen
- displayed the Group to which a job belongs in the Active Environment screen

Recommended Reading

Before continuing with the next chapter, you should read the following in the *CONTROL-M for z/OS User Guide*:

In Chapter 2

- information related to filters and filtering the display in the Active Environment screen
- information about display types of the Active Environment screen

- information about adding a note in the Zoom screen and displaying the note in the Active Environment screen
- commands and options of the Active Environment screen
- the Job Dependency Network screen
- the Job Order Execution History screen, and the Sysout Viewing screen
- status information in the Active Environment screen
- the Statistics screen
- any other information about the Active Environment screen that you have not yet read

Job Ordering and New Day Processing

This chapter includes the following topics:

Overview	194
Preparations	194
Job Ordering Through Utility CTMJOB	194
New Day Processing	197
User Daily Jobs	198
Defining the JCL of a User Daily Job	198
Defining the Date Control Record for a User Daily Job	199
Defining the Job Scheduling Definition of a User Daily Job	201
Additional Points About User Daily Jobs	201
Job Ordering Through Online Utility CTMJOB.RQ	203
Other Methods of Job Ordering	205
Review	206
Recommended Reading	206

Overview

In the previous chapters of this guide, you used the O (Order) or F (Force) option in the Job List screen to order individual jobs. These same options are available in the Table List screen to order or force all the jobs in a particular table.

There are, however, other methods of job ordering. In this chapter of the guide, you will look at job ordering through the following methods:

- Batch utility CTMJOB
- New Day Processing and User Daily jobs
- Online utility CTMJOBQR

You will also see a brief description of several other methods of job ordering.

NOTE



Only relevant DD statements are illustrated in the sample JCLs for the exercises in this chapter. Other DD statements, such as DALOG and DACNDF, are not shown.

Preparations

No special preparations are required for this chapter of the guide.

Job Ordering Through Utility CTMJOB

The CTMJOB job ordering utility can be invoked from the job step or by calling the program from a TSO environment and/or application program.

Assume that you want to order jobs IDJOB1, IDJOB2, and IDJOB3 in table IDGS1.

There are quite a few ways to do this using CTMJOB. Several of these ways are illustrated in the following pages.

NOTE



Specify the appropriate library in place of CTM.TEST.SCHEDULE in each of the alternatives.

1 Define the JCL for CTMJOB as follows:

```
//SCHDLI B DD DSN=CTM. TEST. SCHEDULE , DI SP=SHR
//DAJOB DD *
ORDER DDNAME=SCHDLI B MEM=I DGS1 JOB=I DJOB1 ODATE=ODATE
ORDER DDNAME=SCHDLI B MEM=I DGS1 JOB=I DJOB2 ODATE=ODATE
ORDER DDNAME=SCHDLI B MEM=I DGS1 JOB=I DJOB3 ODATE=ODATE
FORCE
//
```

Notice a few things about this JCL.

- The desired job scheduling definitions are specified in ORDER control statements included in a DAJOB DD statement.
- Each ORDER control statement identifies a library, table and job scheduling definition. To avoid repetition of the scheduling library name in each ORDER statement, the library name was identified in a preceding DD statement.
- The last ORDER statement requested a FORCE. Therefore, job IDJOB3 will be forced regardless of its basic scheduling criteria.

The CTMJOB utility generally uses a DAJOB DD statement to identify the tables and/or jobs to schedule, although an exception will be shown later in this chapter. But the ORDER statements do not have to be included in line in the DAJOB statement. They can, instead, be placed in a parameter member pointed to by the DAJOB statement.

The following steps show how to create an alternative definition for CTMJOB that places the order statements in a PARM member (IDPRM1).

NOTE

Be sure to specify an appropriate PARM library in place of CTM.TEST.PARM.

**2 Define the JCL for CTMJOB as follows:**

```
//SCHDLI B DD DSN=CTM. TEST. SCHEDULE DI SP=SHR
//DAJOB DD DI SP=SHR, DSN=CTM. TEST. PARM(1 DPRM1)
//
```

3 Specify the following contents in the IDPRM1 member in the PARM library:

```
ORDER DDNAME=SCHDLI B MEM=I DGS1 JOB=I DJOB1 ODATE=ODATE
ORDER DDNAME=SCHDLI B MEM=I DGS1 JOB=I DJOB2 ODATE=ODATE
ORDER DDNAME=SCHDLI B MEM=I DGS1 JOB=I DJOB3 ODATE=ODATE
FORCE
```

The above definitions for CTMJOB achieve the same results. The advantage of the second method, using the PARM member, is that you do not have to change the JCL if you want to change the list of jobs to be ordered. You merely need to change the contents of the PARM member.

If the table being scheduled is a Group scheduling table, you can specify scheduling tags to include or ignore in the ORDER statements.

4 Define the following JCL for CTMJOB:

```
//DAJOB DD *
ORDER DDNAME=CTM. TEST. SCHEDULE MEM=I DGS2 ODATE=ODATE
SELECT TAG EXERCISES
IGNORE TAG EOW
IGNORE TAG EOM
//
```

Now that you have seen several instances of using ORDER statements, take a closer look at the ORDER statement syntax.

The syntax for the ORDER statement is as follows:

```
ORDER{ DSN=schedlib | DDNAME=dd | DD=dd } , { MEMBER=table | MEM=table } [ , JOB=jobnm ]
[ , ODATE=date | DATE=date ] [ , ODATEOPT={VALUE | VAL | RUN} ] [ , FORCE ]
[ SELECT TAG tagname1 ]
[ IGNORE TAG tagname2 ]
```

Observe the following points about ORDER statements:

- Immediately following the ORDER statement, you must specify either the name of scheduling library (in a DSN parameter), or the name of a DD statement that identifies a scheduling library (in a DD or DD name parameter).
- You must then specify a table name.
- Jobname is optional. If not specified, all jobs in the table are ordered.
- A date parameter, such as ODATE or DATE, is optional if a date control record has been defined.
- The FORCE parameter is optional. It forces a job even if its basic scheduling criteria are not satisfied.

- IGNORE and SELECT tag statements are relevant for Group scheduling only.

For more information on the parameters in the ORDER statement, see the description of the format of ORDER statements in the CONTROL-M chapter of the *INCONTROL for z/OS Utilities Guide*.

Thus far, all instances of DD statement DAJOB included, or pointed to, a PARM member that includes ORDER statements. However, DAJOB statements can be defined without ORDER statements.

5 Define the following JCL for CTMJOB:

```
//DAJOB      DD  DI SP=SHR, DSN=CTM. TEST. SCHEDULE (I DGS1)
//          DD  DI SP=SHR, DSN=CTM. TEST. SCHEDULE (I DGS3)
//
```

Using this method, the schedule library and table is specified directly in DAJOB DD statement. This method provides no advantage over in-line specification of ORDER statements. It has the disadvantage that you cannot request specific jobs, only whole tables.

It is also possible to use CTMJOB without specifying a DAJOB DD statement. This is done by specifying a PARM parameter in an EXEC statement in the JCL. For example:

```
//ORDERJOB EXEC  PGM=CTMJOB,
//          PARM=' ORDER DSN=CTM. TEST. SCHEDULE TABLE=I DGS1 JOB=SORT ODATE=020201'
//
```

However, this method is of very limited usefulness because it can only be used to order a single table or job.

If you were to submit any of the defined JCLs for CTMJOB, the specified job scheduling definitions would be ordered. However, instead of submitting the JCLs, for this exercise you should move on to the topic of User Daily jobs.

New Day Processing

As mentioned in [Chapter 5, “AutoEdit and JCL,”](#) each site defines a time that represents the end of the old working day and the start of the new working day. At the start of the new working day, New Day processing is performed. New Day processing is the process by which daily maintenance is performed and jobs to be scheduled on the new day are placed in the Active Jobs file.

New Day processing is generally performed by a combination of the New Day procedure and User Daily jobs. The topic of New Day Processing is described in detail in the CONTROL-M chapter of the *INCONTROL for z/OS Administrator Guide*.

You can now take a look at User Daily jobs.

User Daily Jobs

A User Daily job is a job scheduling definition and JCL job that executes the User Daily procedure CTMDAILY. This procedure, in turn, calls the CTMILU program.

The CTMILU program can be seen as an enhanced CTMJOB, in that it executes CTMJOB, but it also executes other programs that provide additional functionality. You will look at one of these other programs shortly, but first, you should create a User Daily job, as described in the following topic.

Defining the JCL of a User Daily Job

- 6 Based on the above JCL, define a User Daily in member IDUDJ1, but do not exit the JCL:

```
//          JCLLIB ORDER=your.proclib. PROCLIB
//          INCLUDE MEMBER=IOASET
//GTSTRT5  EXEC CTMDAILY,
//          DATEREC=DATERECU                <== CHANGE
//DAJOB    DD      *
           ORDER DDNAME=SCHDLIB MEM=IDGS1 JOB=IDJOB1
           ORDER DDNAME=SCHDLIB MEM=IDGS1 JOB=IDJOB2
           ORDER DDNAME=SCHDLIB MEM=IDGS1 JOB=IDJOB3
```

The DAJOB DD statement looks familiar because these parameters are ultimately passed to CTMJOB. Therefore, any syntax that is valid for passing parameters to program CTMJOB, as demonstrated in previous steps in this chapter, can be used in the User Daily job.

But there is a difference. Note that the order statements do not contain date parameters as they did in the previous steps. Instead the date information is provided by a record called the Date Control record. You can now identify the Date Control record to the job, and then create this Date Control record, using the DATEREC parameter in the procedure that points to the member containing the Date Control record.

Now note the statement `DATEREC=DATERCU`. You should examine this item for a moment.

7 In the JCL, change this statement:

```
//          DATEREC=DATERCU
```

to this statement:

```
//          DATEREC=I IDDCRU
```

and then exit the JCL.

Defining the Date Control Record for a User Daily Job

In your PARM library is a member called `DATERECU`. This is the model Date Control record provided with `CONTROL-M`. The sample Date Control record is displayed similar to the following:

```
***** Top of Data *****
000001 301000          301000 301000          301000 301000          301000
***** Bottom of Data *****
```

The Date Control record contains an example date that is repeated several times in specific columns. The integrity of these columns must be maintained. The date must appear in the Date Control record of a User Daily job in the following columns:

- 01 - 06
- 18 - 23
- 25 - 30
- 43 - 48
- 50 - 55
- 67 - 72

Each User Daily requires its own Date Control record. You will use this model record to create a Date Control record for the User Daily that you have just created. As you can see from the step you just performed, you are going to call the Date Control record for this User Daily `IDDCRU`.

8 In the PARM library, copy member `DATERECU` and call it `IDDCRU`.

9 Edit member `IDDCRU` and, without changing the column positions, replace the old dates with the current working date. Then exit the member.

Once you have created a Date Control record for a User Daily, CONTROL-M automatically updates the date information in the record. This is one of the great advantage of using User Dailies to order jobs. You do not have to update date information in a DAJOB DD statement each day, because CONTROL-M provides the updated information through the Date Control record.

This appearance of the date six times (instead of once) in the Date Control record of a User Daily enables CONTROL-M to manage the process of job ordering. At each stage in the job ordering process, the current original scheduling date is placed in one of these fields.

Enhanced Daily Checkpointing

An optional, second, Date Control record can be defined for a User Daily to enable Enhanced Daily Checkpointing. If an interruption such as a system crash occurs during job ordering, Enhanced Daily Checkpointing facilitates automatic rerun of the job ordering process by enabling CONTROL-M to identify the last job ordered before the interruption. CONTROL-M can then continue the ordering process from that point.

This second Date Control record has a completely different format than the regular Date Control record. In fact, the Enhanced Daily Checkpointing Date Control record contains far more than dates.

The following table shows the format of the second Date Control record, which is required only if Enhanced Daily Checkpointing is used.

Table 21 Format of the Second Date Control Record (For Enhanced Daily Checkpointing Only)

Column	Constant or Value Added	Description
01 - 04	JOB=	Constant.
05 - 12	blank	In this area, CONTROL-M stores the MEMNAME value of the last ordered job.
13 - 23	,SERIAL_NO=	Constant (note the comma before the "S").
24 - 28	blank	In this area, CONTROL-M stores its internal sequence number of the last ordered job.
29 - 37	,ORDERID=	Constant (note the comma before the "O").
38 - 42	blank	In this area, CONTROL-M stores the order ID of the last ordered job.
43 - 49	,GROUP=	Constant (note the comma before the "G").
50 - 69	blank	In this area, CONTROL-M stores the group name of the last ordered job.

When creating this record, you must

- specify the indicated constants, such as JOB=, in the appropriate columns
- leave blank the columns indicated as blank

These columns are filled in by CONTROL-M during User Daily processing.

Defining the Job Scheduling Definition of a User Daily Job

You have now defined the JCL of User Daily job IDUDJ1, and created its Date Control record called IDDCRU. One task remains, however; you must define the job scheduling definition for User Daily IDUDJ1.

- 10** In your scheduling library, create a new table called IDUDT1, and in it create a job scheduling definition called IDUDJ1. Define the job scheduling definition as follows:
 - A** Ensure that the MEMNAME and MEMLIB fields point to the JCL of IDUTJ1.
 - B** Define the Basic Scheduling parameters so that this job scheduling definition gets ordered every day, for example, by using a DAYS value of ALL.
 - C** Exit the job scheduling definition.

Additional Points About User Daily Jobs

As you saw in previous chapters you can order jobs in the online facility. However, using this method, you must first enter the online facility. Then you must order the jobs, either individually or a table at a time. This method is not bad for an occasional job order, but it certainly is not suitable for ordering a large number of jobs, especially if they are from different scheduling tables.

In this chapter, you saw that you can order a large number of jobs using a single run of CTMJOB. And you can keep the list of jobs you specified to CTMJOB for reuse. This eliminates the disadvantages of using online job ordering on regular daily basis. |

For daily processing, employing User Daily jobs provides an additional advantage—the Date Control record. The Date Control record has many advantages, including the following facts:

- It can be used in conjunction with the RETRO parameter, described in the parameters chapter of the *CONTROL-M for z/OS User Guide*, to enable the site to catch up with missed work days—for example, if the system is down for several days.
- It enables Enhanced Daily Checkpointing, which is useful in case of system crash.

However, even the User Daily job must be ordered. There are several ways this can be done:

- You can order the User Daily job using the Online facility. But this requires going into the CONTROL-M Online facility—still a manual intervention.
- You can order the User Daily job using CTMJOB.
- You can use any the other job ordering methods available, some of which will be described below. But this, too, involves manual intervention.
- Finally, you can order User Dailies using the New Day procedure. The New Day procedure is defined and maintained by the CONTROL-M administrator, and is run once each day as part of New Day processing. When User Dailies are defined to the New Day procedure, they are automatically ordered each day that their basic scheduling criteria are satisfied. Most sites use a combination of User Daily jobs and the New Day procedure to completely automate daily job scheduling.

Because the New Day procedure is defined by the CONTROL-M administrator, and is discussed in detail in the *INCONTROL for z/OS Administrator Guide*, it is not discussed in this guide.

Before moving on to a discussion of alternative methods of job ordering, you should note the following points about User Daily jobs:

- CONTROL-M provides two sample User Daily job scheduling definitions in member MAINDAY in the SCHEDULE library:
 - DAILYSYS is a sample User Daily for scheduling system jobs
 - DAILYPRD is a sample User Daily for scheduling production jobs.
- You can define as many User Daily jobs as you want, with each ordering only those jobs that you want it to order. This leaves you great flexibility in organizing your User Daily jobs by whatever system is useful. For example, you can organize User Dailies according to scheduling table, application or group, department, basic and runtime criteria of the jobs being ordered, or any other useful criteria or combination of criteria.

- You can define the scheduling criteria of the User Daily job in any way you wish. For example, if a certain set of jobs is to be processed at the end of the work day, there is no need for them to sit in the Active Jobs file all day. Instead, you can ensure that they are ordered only in the evening or at night, by appropriately defining the TIME FROM criteria of the User Daily that orders those jobs.
- If you want, you can define a User Daily to run several maintenance procedures that you would like run in succession.
- Although the New Day procedure is ordered only once each day, at start of New Day, you can order User Daily jobs whenever you wish, and as often as you wish. Of course, you must ensure that this does not produce unwanted results. If you do not want multiple orders of the same job, you should not run the User Daily more than once.
- You can locate your User Daily jobs as you wish. For example, if all the User Daily jobs are placed in a single table, then by ordering that table you order all User Daily jobs that it contains.

Job Ordering Through Online Utility CTMJOBRO

As previously discussed, the occasional job scheduling request can be made using the Order or Force option in the Scheduling Definition facility. At some sites, however, access to Scheduling Definition facility may be restricted for security reasons, for example, to prevent the changing or deletion of job scheduling definitions.

As an alternative to ordering or forcing a job in the Scheduling Definition facility, you can order or force a job using Online Utility M1.

- 11** Enter the Online Utilities menu and request utility CTMJOBRO, which is Option M1. The Job Request Utility panel is displayed.

```

----- JOB REQUEST UTILITY -----
COMMAND ==>

SCHEDULING LIBRARY    ==> CTM.TEST.SCHEDULE
TABLE NAME            ==>
JOB NAME              ==> (* for all jobs)
SCHEDULED RUN DATE    ==> 02 02 01 (ODATE - format MM DD YY)
FORCED SCHEDULING     ==> NO (YES, NO)
ENTER YES TO CONTINUE ==>
GROUP                 ==> (Optional)

CALENDAR LIBRARY      ==> IOA.TEST.CAL
    
```

In this panel you can

- specify the name of the library, table and job scheduling definition

To request all jobs in a table, specify an asterisk in the job name field.

- specify the scheduling run date

The basic scheduling criteria of the jobs are checked against this date to see if the job should be ordered as requested.

- indicate if you want the job forced if it should not be scheduled on that particular run date
- specify the name of the calendar library

This tells CONTROL-M where to look if a calendar name is specified in the job scheduling definition. Calendars are discussed in [Chapter 8, “Additional Features.”](#)

- specify a group name

NOTE



The GROUP field is generally useful only in regular scheduling tables, not Group tables, and only if an * is specified in the JOB NAME field. It limits the jobs ordered to those belonging to the specified group.

In general, it is probably quicker to use batch utility CTMJOB, especially if you are ordering several jobs. Perhaps the only advantage of online utility CTMJOB_{RQ} is that you do not need to know the format of the order statements—you merely fill in the parameter fields presented in the panel. But CTMJOB_{RQ} is definitely not a power utility.

Other Methods of Job Ordering

Thus far, you have seen the following methods of job ordering:

- Order and Force options in the Job and Table List screens
- Batch utility CTMJOB
- User Daily jobs
- Online utility CTMJOB_{RQ}

The following job ordering methods are also available:

Table 22 Non-Routine Job Ordering Methods

Method	Description
End User Job Order interface	Enables job ordering through online utility (or CLIST) CTMJOB _{RQ} . Described in the <i>CONTROL-M for z/OS User Guide</i> .
Utility CTM _{BLT}	Described in the CONTROL-M chapter of the <i>INCONTROL for z/OS Utilities Guide</i> .
CLIST CTM _{CJOBS}	Enables job ordering directly from the TSO environment. Described in the <i>CONTROL-M for z/OS User Guide</i> .
Quick submit command CTM _{QSB}	Enables job ordering through CONTROL-M submit command CTM _{QSB} instead of the ISPF submit command. Described in the <i>CONTROL-M for z/OS User Guide</i> .
Job ordering from special environments	Facilitates job ordering from other environments, such as CICS and ROSCOE, through CTM _{AJO} . Described in the <i>CONTROL-M for z/OS User Guide</i> .

12 Exit the Online facility.

This completes the steps in this chapter of the *CONTROL-M for z/OS Getting Started Guide*.

Review

In this chapter you

- learned to order jobs using batch utility CTMJOB, and learned several different methods and formats that can be used to let the utility know which jobs to order
- learned to define the JCL and job scheduling definition for a User Daily job, and how to define its required Date Control record, and a second, optional Date Control record for Enhanced Daily Checkpointing
- took a look at the CTMJOBQR online utility panel that can be used to order a job
- briefly identified several other methods of job ordering

Recommended Reading

Before continuing with the next chapter, it is recommended that you read the following:

- In the *CONTROL-M for z/OS User Guide*:
 - In Chapter 2, the description of online utility CTMJOBQR (M1).
 - In Chapter 3, the description of the RETRO parameter.
 - In Chapter 6, the topic “Job Ordering Methods.”
- In Chapter 3 of the *CONTROL-M for z/OS Administrator Guide*, the topic “Job Ordering using New Day Processing.”
- In Chapter 3 of the *CONTROL-M for z/OS Utilities Guide*, the description of utility CTMJOB.

Additional Features

This chapter includes the following topics:

Overview	208
Preparations	208
Defining Calendars	209
Structure of the IOA Calendar Definition Facility	214
Responding to External Events through CMEM	215
Event Types Handled by CMEM - Available ON Statements	221
Creating On Spool Jobs	222
Defining On Spool Jobs	223
Bringing the Job On Spool	227
Additional Points About On Spool Jobs	228
Batch Utility IOACND	229
Miscellaneous Facilities	232
History Jobs File	232
Journaling and Restoration Capability	232
Accumulating Statistics: Statistics Facility	233
Automatic Tape Adjustment	234
Simulating Job Execution and Forecasting Resource Usage	234
CONTROL-M/Restart Dataset Cleanup Utility CTRCCLN (R2)	235
Reporting Facility	235
Exit the Online Session	235
Review	236
Recommended Reading	237

Overview

In this, the final chapter of the Getting Started guide, you will examine several topics not covered in the other chapters.

What you learn in this chapter will enable you to

- create a calendar in the IOA Calendar facility
- take a look at, and create a rule in, the CMEM (CONTROL-M Event Manager) facility
- bring a job On Spool in CMEM
- use batch utility IOACND to clean up conditions and resources

You will also learn about the availability of several miscellaneous facilities.

Preparations

To prepare for the exercises in this chapter

- create a new scheduling table (a regular, not a group, table) called IDGS4
- create a new job scheduling definition in table IDGS4, called IDJOB10, with these characteristics:
 - In the MEMNAME field, specify **IDJOB10**
 - In the GROUP field, specify **IDGRP5**
 - In the DESCRIPTION field, specify **THIS IS JOB10**
 - In the OUT field, specify **IDJOB10-ENDED-OK ODAT +**
- create a trivial JCL for IDJOB10
- Create a JCL called IDEXT1

The JCL should contain two steps, S1 and S2, each ending with condition code C0000. There should be no job scheduling definition for this JCL.

Defining Calendars

A calendar definition is a collection of parameters, in calendar form, that is used to indicate on what dates of the year scheduling should occur.

Calendars can be very useful for providing basic scheduling criteria, especially when

- scheduling dates do not follow a pattern, or follow a pattern that is not easily specified using regular basic scheduling criteria

An example might be a WORKDAYS calendar that schedules jobs on all weekdays, Monday through Friday, that are not holidays. Creating this schedule using basic scheduling parameters would be problematic. Defining the weekdays is simple enough, but excluding the holidays will be a problem.

- identical detailed and complex scheduling criteria are used for many jobs

Instead of detailing the scheduling criteria in the job scheduling definition of each job, the scheduling criteria can be specified once in a calendar, and then only the calendar name need be specified in each the job scheduling definition. An example might be a HOLIDAYS calendar that consists of a number of disparate dates. Specifying this once in a calendar is easier and less error-prone than specifying the same set of dates in many job scheduling definitions.

You can now define a calendar for use in your job scheduling definition.

- 1 Enter =8 in the COMMAND field. The IOA Calendar Definition Facility entry panel is displayed.

In the LIBRARY field is the name of the default calendar library at your site. For purposes of illustration, you will call this library IOA.TEST.CAL.

Figure 51 IOA Calendar Definition Facility - Entry Panel

```
----- IOA CALENDAR FACILITY - ENTRY PANEL -----(8)
COMMAND ==>>

SPECIFY LIBRARY, CALENDAR, YEAR

LIBRARY ==>> IOA.TEST.CAL
CALENDAR ==>> (Blank for calendar selection list)
YEAR ==>> (Blank for year selection list)

USE THE COMMAND "SHPF" TO SEE PFK ASSIGNMENT 11.28.41
```

2 In the CALENDAR field, enter the name IDCAL1 and press Enter.

```
SPECIFY LIBRARY, CALENDAR, YEAR

LIBRARY ==>> IOA.TEST.CAL
CALENDAR ==>> IDCAL1 (Blank for calendar selection list)
YEAR ==>> (Blank for year selection list)
```

The Calendar Definition screen is displayed for the current year, as indicated by the label under the COMMAND field.

Figure 52 IOA Calendar Definition Screen

```

----- IOA CALENDAR - IDCAL1 -----(8. Y)
COMMAND ==>                               SCROLL==> CRSR
YEAR 2001 -

----S-----S-----S-----S-----S-----S----
   1 2 3 4 5 6 7 8 9 + 1 2 3 4 5 6 7 8 9 + 1 2 3 4 5 6 7 8 9 + 1
01
----S-----S-----S-----S-----S-----S----
   1 2 3 4 5 6 7 8 9 + 1 2 3 4 5 6 7 8 9 + 1 2 3 4 5 6 7 8
02
----S-----S-----S-----S-----S-----S----
   1 2 3 4 5 6 7 8 9 + 1 2 3 4 5 6 7 8 9 + 1 2 3 4 5 6 7 8 9 + 1
03
----S-----S-----S-----S-----S-----S----
   1 2 3 4 5 6 7 8 9 + 1 2 3 4 5 6 7 8 9 + 1 2 3 4 5 6 7 8 9 +
04
----S-----S-----S-----S-----S-----S----
   1 2 3 4 5 6 7 8 9 + 1 2 3 4 5 6 7 8 9 + 1 2 3 4 5 6 7 8 9 + 1
05
----S-----S-----S-----S-----S-----S----
   1 2 3 4 5 6 7 8 9 + 1 2 3 4 5 6 7 8 9 + 1 2 3 4 5 6 7 8 9 +
06
----S-----S-----S-----S-----S-----S----
TYPE Y IN ALL THE EXECUTION DAYS                               11. 35. 33

```

3 Tab to the field next to the year label, and specify **GETTING STARTED**. The following screen is displayed:

```

----- IOA CALENDAR - IDCAL1 -----(8. Y)
COMMAND ==>                               SCROLL==> CRSR
YEAR 2001 - GETTING STARTED

```

Notice the layout of the screen:

- It is sectioned according to months. Only the first six months are displayed. You can page down to display the next six months.
- Each section contains three lines:
 - A number line contains up to 31 digits indicating the dates in the month. A plus sign indicates the 10th, 20th and 30th.
 - Directly below the number line is a blank line used for entering a Y value for each date on which scheduling should occur.
 - Directly above the number line is a line containing Ss. Each S indicates either Saturday or Sunday, depending on the defaults defined at your site.

Normally, you would try to create a logical, useful calendar. However, to keep the definition as simple as possible, you will specify Y only for the date on which you are performing these exercises.

- 4 Specify Y in the field that represents the current working date. The Calendar definition should have only one entry:

```

----- IOA CALENDAR - IDCAL1 -----(8.Y)
COMMAND ==>                                SCROLL==> CRSR
YEAR 2001 - GETTING STARTED

-----S-----S-----S-----S-----S-----S-----S---
      1 2 3 4 5 6 7 8 9 + 1 2 3 4 5 6 7 8 9 + 1 2 3 4 5 6 7 8 9 + 1
01
-----S-----S-----S-----S-----S-----S-----S---
      1 2 3 4 5 6 7 8 9 + 1 2 3 4 5 6 7 8 9 + 1 2 3 4 5 6 7 8
02
-----S-----S-----S-----S-----S-----S-----S---
      1 2 3 4 5 6 7 8 9 + 1 2 3 4 5 6 7 8 9 + 1 2 3 4 5 6 7 8 9 + 1
03
-----S-----S-----S-----S-----S-----S-----S---
      1 2 3 4 5 6 7 8 9 + 1 2 3 4 5 6 7 8 9 + 1 2 3 4 5 6 7 8 9 +
04
-----S-----S-----S-----S-----S-----S-----S---
      1 2 3 4 5 6 7 8 9 + 1 2 3 4 5 6 7 8 9 + 1 2 3 4 5 6 7 8 9 + 1
05
      Y
-----S-----S-----S-----S-----S-----S-----S---
      1 2 3 4 5 6 7 8 9 + 1 2 3 4 5 6 7 8 9 + 1 2 3 4 5 6 7 8 9 +
06
-----S-----S-----S-----S-----S-----S-----S---
TYPE Y IN ALL THE EXECUTION DAYS                                11. 35. 33
    
```

- 5 Exit the Calendar Definition screen. The Year List screen is displayed, listing only the definition you just created.

Structure of the IOA Calendar Definition Facility

Before continuing with the exercises, you should note some of the similarities between the structure of the Scheduling Definition facility and the structure of the Calendar Definition facility.

- Both job scheduling definitions and calendar definitions are stored in members in a library. Therefore, both facilities have corresponding screens at each relevant level.
 - The gateway to both definition facilities is an entry panel that allows specification of a Library, member, and entity, either a calendar year or a job scheduling definition name.
 - The screen at the next level lists the members in the library
 - The Table List screen lists the members in the Scheduling Definition facility.
 - The Calendar List screen lists the members in the Calendar facility.
 - The screen at the next level lists the entities in the member.
 - The Job List screen lists the entities in the Scheduling Definition facility member.
 - The Year List screen lists the entities in the Calendar facility member.
 - The screen at the next level is the definition screen itself.
 - The Job Scheduling Definition screen provides Scheduling Definition facility information.
 - The Calendar Definition screen provides Calendar Definition facility information.
- The description you provide in the definition screen appears in the Job List screen or the Year List screen.
- If you create or modify a definition, an Exit Option window is displayed upon exiting the Job List screen or Year List screen to enable you to save or cancel your work.

Responding to External Events through CMEM

Until now, almost everything you have done in this guide has been connected to the scheduling of jobs under the CONTROL-M monitor. You learned how to define jobs, how to order the jobs, how to check up on, and intervene in, the processing, how to perform restart, and so on.

However, CONTROL-M provides a facility, the CONTROL-M Event Manager (CMEM) facility, that enables CONTROL-M to perform specified actions in response to external events. External events are events in the system that occur outside direct operation of CONTROL-M, such as submission of a job not under the control of the CONTROL-M monitor.

The CMEM facility is comprised of a monitor and a subsystem. The facility employs sets of user-defined rules that specify events to monitor and actions to perform if a specified event occurs.

You can now enter the CMEM facility.

- 9 Enter Option C in the IOA Primary Option menu, or =C from any other location. The CMEM entry panel is displayed.

Figure 54 CMEM Rule Definition Facility – Entry Panel

```

----- CMEM RULE DEFINITION FACILITY - ENTRY PANEL -----(C)
COMMAND ===>

SPECIFY LIBRARY, TABLE NAME, RULE NAME

LIBRARY   ===> CMEM.TEST.RULES
TABLE     ===>                               (Blank for table selection list)
RULE      ===>                               (Blank for rule selection list)

USE THE COMMAND SHPF TO SEE PFK ASSIGNMENT                                22.35.51

```

CMEM has a library, member, and rule structure much like the library, member, and job scheduling definition structure of the Scheduling Definition facility.

This is reflected in the similarity between the Scheduling Definition entry panel and the CMEM entry panel, which has entry fields for LIBRARY, TABLE, and RULE.

Other screens of the CMEM facility also correspond to those of the Scheduling Definition facility:

- Table List Screen
- Rule List screen, which corresponds to the Job List screen
- Rule Definition screen, which corresponds to the Job Scheduling Definition screen

NOTE



Many commands and options in the corresponding screens are also the same.

You can now create a CMEM table and rule. Use a test CMEM library if one exists, or use the default CMEM rule library. For the purposes of this guide, you will call this library CMEM.TEST.RULES.

- 10 In the CMEM entry panel, enter **CIDM1** in the TABLE field, and rule **IDRUL1** in the RULE field. Use either the default or a test CMEM rule library. For the purposes of this guide, this library is called CMEM.TEST.RULE.

NOTE



If CONTROL-O is installed, your site should not be running CMEM.

```

SPECIFY LIBRARY, TABLE NAME, RULE NAME

LIBRARY    ==> CMEM. TEST. RULES
TABLE      ==> IDC1          (Blank for table selection list)
RULE       ==> IDRUL1       (Blank for rule selection list)
    
```

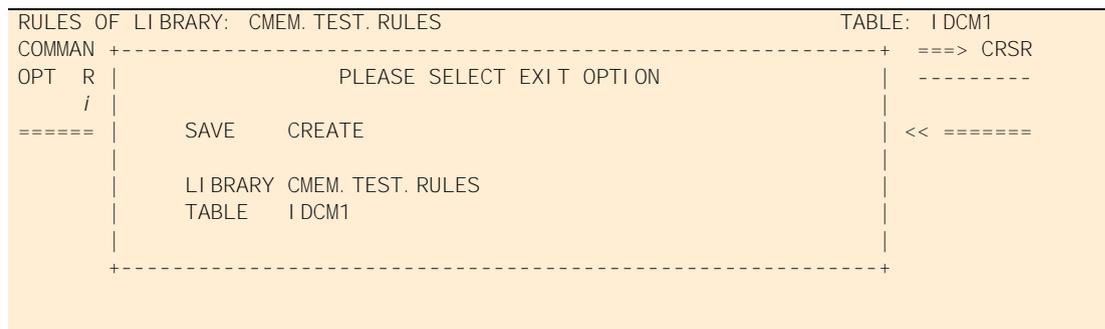
The Rule Definition is displayed. It immediately displays the following message at the top of the screen:

```

IOAE4CE AT LEAST ONE "ON" STATEMENT MUST BE FILLED IN
    
```


17 Exit the Rule List screen. The Exit Option window is displayed.

Figure 57 CMEM Exit Option Window



18 Enter **Y** in the **CREATE** field. The CMEM entry panel is displayed again.

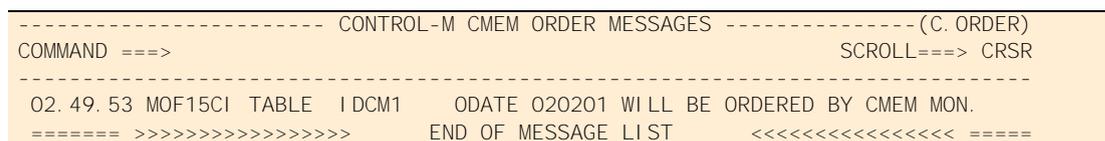
You can now load to memory the table that contains the rule you just defined.

19 Ensure the **TABLE** and **RULE** name fields are blank and press **Enter**. The Table List screen is displayed.

20 Enter Option **F** (Force) for the table **IDCM1**. If a Confirmation window is displayed, confirm the request.

The **CONTROL-M CMEM Order Messages** screen displays a message indicating that the table is about to be ordered by the CMEM monitor.:

Figure 58 CONTROL-M CMEM Order Messages Screen



21 Exit the **CONTROL-M CMEM Order Messages** screen. The **CMEM Table List** screen is displayed.

22 Enter **=4** to check the **IOA Conditions/Resources** screen for condition **IDS2-IDEXT1-OK**.

The condition should not appear because the rule that creates it has not yet been triggered.

23 Submit the **JCL** of job **IDEXT1**. Once **Step S2** has ended **OK**, check again for condition **IDS2-IDEXT1-OK** in the **IOA Conditions/Resources** screen.

The condition does not appear, due to the rule being in TEST mode, as specified in [13 on page 218](#). However, if the rule had been defined in PROD mode, the condition would appear, indicating that the CMEM rule detected the successful completion of step S2 and added the specified condition to the IOA Conditions file. CONTROL-M jobs that have this condition as an IN condition can now be submitted, assuming all other scheduling and runtime criteria are satisfied.

Event Types Handled by CMEM - Available ON Statements

You have already seen the ON STEP event. Shown below is a table with the complete list of ON statements available to CMEM.

Table 23 ON Statements Available to CMEM

Event	Description
DSNEVENT	Dataset disposition, whether cataloged, deleted, or kept, during step termination or dynamic decollation, or the occurrence of a NOT CATLGD 2 event, which usually occurs when a dataset name is created in a job step but not cataloged because its name already exists in the MVS catalog. Specified in an ON DSNEVENT statement in the rule.
JOBARRIV	Arrival of a job on the JES spool from any source. Examples Jobs submitted by a TSO user or by CICS, or jobs received over an NJE network. Specified in an ON JOBARRIV statement in the rule.
JOBEND	Completion of a job regardless of its source. Specified in an ON JOBEND statement in the rule.
STEP	Termination of a job step. Specified in an ON STEP statement in the rule.

The following list indicates the Rule type code that appears in the Rule List screen, depending on the type of ON statements specified in the rule:

- R – Job arrival
- X – Job end
- D – Dataset
- Z – Step

CMEM Actions - Available DO Statements

You have already used a DO COND statement in your rule. The following table provides the complete list of DO statements available to CMEM:

Table 24 CMEM Action Parameters

Parameter	Description
DO statement	<p>Action to be performed when the rule is triggered. Subparameters may be displayed. Valid DO statements are:</p> <ul style="list-style-type: none"> ■ DO COND—Add or delete a prerequisite condition. ■ DO FORCEJOB—Force a job order under CONTROL-M. ■ DO STOPJOB—Stop execution of the remaining steps of the job that triggered the rule. <p>The following actions can be defined if CONTROL-O is installed:</p> <ul style="list-style-type: none"> ■ DO RULE—Invoke a CONTROL-O rule from within the current rule. ■ DO SHOUT—Issue a message to a specified destination using the Shout facility.

Creating On Spool Jobs

An On Spool job is an independently submitted job or started task, such as a job submitted by a TSO user, that is brought under the control of the CONTROL-M monitor using a CMEM rule.

Such a CMEM rule must

- be an ON JOBARRIV rule that is triggered by the arrival of the job on the JES spool
- have a DO FORCEJOB statement that forces a job scheduling definition that “matches” the arriving job, so that CONTROL-M can use the instructions in the job scheduling definition to control the job

The issue of matching the job scheduling definition to the arriving job is discussed in more detail later.

CONTROL-M then controls the entire life cycle of the job, except submission, according to the instructions in the forced job scheduling definition. Because the job has already been submitted, CONTROL-M does not submit the job. However, if the job is held, CONTROL-M releases the job when the runtime scheduling criteria are met.

Defining On Spool Jobs

The following components are necessary to create On Spool jobs:

- job to be brought On Spool
- CMEM rule
- job scheduling definition

The following tables clarify the connections between the components by identifying the values you will specify and the reasons for those values. This understanding should be gained before you create the necessary components.

Table 25 Job To Be Brought On Spool - Values for Exercise

Item	Value	Description
Member	IDEXT2	Name of the JCL member of the External job.
Library	“CTM.TEST. JCL”	The test (or personal) JCL library you have been using for these exercises. As in the previous exercises, call it CTM.TEST.JCL

Table 26 CMEM Rule - Values for Exercise

Item	Value	Description
Rule	IDEXP2	Name of rule that will bring job IDEXT2 On Spool.
Member	IDCM1	Rule Table that contains the rule, which is shown in the same table as in previous exercise.
Library	“CMEM.TEST. RULE”	CMEM Rule library used in previous CMEM exercise.
ON	ON JOBARRIV IDEXT2	On statement in rule that detects the arrival of the external job to be brought on spool.
DO	FORCEJOB	Statement that forces the job scheduling definition that matches the arrived job IDEXT2. TABLE, JOB, and LIBRARY are the relevant subparameters of this statement, and are described below:
TABLE	IDGS4	Table that contains the job scheduling definition.
JOB	IDEXT2	Name of the job scheduling definition that matches the arrived job. If left blank, CONTROL-M will search the specified table for the matching job.
LIBRARY	“CTM.TEST. SCHEDULE”	Scheduling library that contains the job scheduling definition.

Table 27 Job Scheduling Definition - Values for Exercise

Item	Value	Description
Jobname	IDEXT2	Name of job scheduling definition. It must match the name of the arrived job.
Member	IDGS4	Name of the scheduling table of the job scheduling definition. The table name specified in DO FORCEJOB statement in the rule must point to this table.
Library	"CTM.TEST. SCHEDULE"	Scheduling library of the job scheduling definition. The library name specified in DO FORCEJOB statement in the rule must point to this table.
MEMNAME field	IDEXT2	Must point to the JCL member of the arrived job.
MEMLIB field	"CTM.TEST. JCL"	Must point to the JCL library of the arrived job.

You can now create the components.

Creating the JCL of the Job

24 Using the values in the preceding tables, for the JCL job, create the JCL of the job that will be placed on the JES Spool. The job can be a trivial job, but ensure that the job card has the following characteristics:

- The job should be submitted with TYPRUN=HOLD to delay its execution and permit CONTROL-M to determine when to run the job.
- The MSGCLASS sysout of the job should be one of the following to enable CONTROL-M to read the sysout for the job and perform postprocessing according to the job scheduling definition:
 - For JES3 users, it must be equal to the CONTROL-M SYSOUT held class.
 - For JES2 users, it can be any held SYSOUT class.

Creating the CMEM Rule

25 Using the values in the preceding tables, for the CMEM rule, create CMEM rule IDEXP2. Ensure that the rule has the following characteristics:

- The ON JOBARRIV statement should have the job name value of IDEXT2 after the = symbol. The rest of the fields can be blank.

Creating the Job Scheduling Definition

27 Using the values in the preceding tables, for the job scheduling definition, create the job scheduling definition **IDEXT2** in table **IDGS4**. Ensure that the definition has the following characteristics:

- The **MEMNAME** field contains **IDEXT2**.
- The **MEMLIB** field points to the JCL library of the arrived job, which in this exercise is **CTM.TEST.JCL**.

You can now define the following parameters in this job scheduling definition:

28 In the **GROUP** field, specify **IDGRP6**.

29 In the **DESC** field, **ON SPOOL EXERCISE**.

30 Fill in the **OUT** fields with the following values:
IDEXT2-ENDED-OK ODAT +

The job scheduling definition is displayed as follows:

The CMEM rule detects the arrival of job IDEXT2 on the JES spool, and forces the job scheduling definition of job IDEXT2. CONTROL-M then recognizes this as an On Spool job and tracks and controls the job using the instructions in the job scheduling definition.

- 34 Check the job log. The job log indicates the appropriate CONTROL-M handling of the job, depending on the results of the execution (assuming that the rule is in PROD mode).
- 35 Check the IOA Conditions/Resources screen.

The condition IDEXT2-ENDED-OK is displayed in the screen. This indicates that the job was brought under the control of the CONTROL-M monitor.

Additional Points About On Spool Jobs

The following points relate to component definition:

- The job name specified in the ON JOBARRIV statement in this rule must match the name of the job to be monitored. It can be a full job name, or it can be a mask if a group of jobs is to be monitored.
- The job scheduling definition must be forced by the first DO FORCEJOB statement in the CMEM rule.
- The MEMNAME value in the job scheduling definition must match the name of the external job. A mask can be specified in the MEMNAME field if the same job scheduling definition is used for more than one job.
- Appropriate runtime scheduling criteria for the job must be defined in the job scheduling definition. This enables CONTROL-M to control when the job should be run, that is, its execution.
- Desired post-processing actions must be defined in the job scheduling definition.

On Spool jobs are handled as follows:

- When the job arrival event occurs, CONTROL-M forces the requested table or job.
 - If the MEMNAME value in the requested table or job does not match the name of the arriving job, the table or job is forced and processed regularly by CONTROL-M, which presumes that a job is submitted when its runtime scheduling criteria are met, and so on.
 - If the MEMNAME value in the requested table or job matches the name of the arriving job, the job becomes an On Spool job and CONTROL-M

- replaces the MEMNAME mask, if a mask was specified in MEMNAME, with the name of the arriving job
- assigns the job ID of the job that triggered the event to the forced job
- forces the job

For details and exceptions see the discussion of On Spool job scheduling definition considerations in the *CONTROL-M for z/OS User Guide*.

The forced job appears in the Active Environment screen with WAIT SCHEDULE ON SPOOL as its status.

- CONTROL-M starts processing the forced job when all runtime scheduling criteria defined in the job scheduling definition are satisfied. If there are no runtime scheduling criteria in the job scheduling definition, CONTROL-M starts processing the job immediately.
- CONTROL-M looks for the job in the spool and releases it, if required.
 - If the external job is waiting for execution in HELD state, that is, if the job arrives on spool with the TYPRUN parameter set to HOLD, CONTROL-M releases it for execution.
 - Otherwise, CONTROL-M verifies that the job is still in the spool, and is waiting for execution, executing, or ended, before performing postprocessing.
- CONTROL-M waits for the job to finish execution, reads its SYSOUT, analyzes the execution results, and performs all the postprocessing actions defined in the job scheduling definition.

By default, CONTROL-M can only handle On Spool jobs that originate on the same NJE node on which CONTROL-M is running.

Batch Utility IOACND

Utility IOACND is a utility that is generally included in New Day processing but can also be run throughout the day on an as-needed basis. The utility adds, verifies the existence of, deletes, or modifies prerequisite conditions and resources in the IOA Conditions file and/or the CONTROL-M Resources file. It can be activated as a started task (STC), from TSO, or from within a user program.

A copy of the utility is pre-supplied with CONTROL-M. You need only provide the desired control statements, as discussed below:

- 36** In the pre-supplied copy of the utility, add the following control statements under the EXEC statement:

```
ADD      COND  I DGS-MONI TOR-UP  %%OMONTH. %%ODAY
DELETE  COND  I DJOB10-ENDED-OK  %%OMONTH. %%ODAY
CHANGE  RESOURCE I DTAPE 3
ADD CONTROL I DDATA1 E
```

This job must be submitted through CONTROL-M to resolve the AutoEdit variables.

- 37** Submit the utility.

- 38** Enter the IOA Conditions/Resources screen.

- Condition IDGS-MONITOR UP appears with the current ODATE.
- Condition IDJOB10-ENDED-OK with the current ODATE no longer appears.
- Three units of quantitative resource IDTAPE appear in the screen.
- Control resource IDDATA1 has been assigned Exclusive control.

Take a closer look at the control statements you can use in this utility. All control statements must have the following format:

```
action entity_type entity_name qualifiers
```

where

- *action* can be one of the following:
 - ADD Valid for any entity type.
 - DELETE Valid for any entity type.
 - CHECK Valid for COND only. Verifies that the condition exists.
 - CHANGE Valid for RESOURCE only. Changes the resource quantity.
- *entity_type* can be one of the following:
 - COND For prerequisite conditions
 - RESOURCE For quantitative resources
 - CONTROL For control resources
- *entity_name* is the name of the condition or resource.
- *qualifiers* depend on the entity type, as indicated in the following table:

Table 28 Qualifiers and Entity Types

Entity Type	Qualifier	Description and Valid Values
COND	cond_date	Condition date. Valid values are: literal – mmdd or ddmm site format DATE – Current Gregorian date WDATE – IOA working date STAT – static date.
RESOURCE	resc_quantity	Resource quantity. In either of two formats: n (integer, such as 2 or 3). Valid for ADD and DELETE actions. +/- n (such as -2 or +3). Valid for CHANGE actions. It specifies the amount and direction of the change.
CONTROL	control_type	Type of Control. Valid values are: E – Exclusive S – Shared

Note the following points about utility IOACND:

- It can be activated as a started task (STC), from TSO, or from within a user program.
- The addition, deletion, modification or verification is recorded three ways:
 - in the job sysout
 - as a message on the operator console
 - as an event in the IOA Log file.
- Parameters can be received by the utility in two ways:
 - using DD statement DACNDIN (or SYSIN)
 - using PARM.

The parameters required depend on whether prerequisite conditions, control resources or quantitative resources are being processed.

- If it is necessary to add, delete, verify, or modify prerequisite conditions or resources during a job step, CTMCND can be called as a procedure from within the application program.

IOA Online Utility I1, which is described in the IOA Utilities chapter of the *CONTROL-M for z/OS User Guide*, can be used for adding, deleting or checking a prerequisite condition in the IOA Conditions file.

Miscellaneous Facilities

The following facilities may prove of special interest to you. There are no exercises to perform regarding these facilities, and you can find their descriptions in the appropriate guide. They are listed below to ensure you know they are available.

Some of the following descriptions are taken from the introductory chapter of the *CONTROL-M for z/OS User Guide*.

History Jobs File

During New Day processing, jobs that have ended OK or whose retention period has expired, according to job scheduling definition parameters, are deleted from the Active Jobs file.

If CONTROL-M/Restart is installed, these jobs can be placed in the History Jobs file during New Day processing. This is an optional feature that can be activated by the INCONTROL administrator. Activation of this feature is described under parameter HIST in the CONTROL-M chapter of the *INCONTROL for z/OS Administrator Guide*.

Jobs in the History Jobs file can be restored, by request, to the Active Jobs file, for subsequent restart.

Jobs remain in the History Jobs file until they are deleted according to criteria defined in the job scheduling definition.

The contents of the History Jobs file can be viewed from the History Environment screen, as described in the online facilities chapter of the *CONTROL-M for z/OS User Guide*.

Journaling and Restoration Capability

The CONTROL-M Journal file collects data about changes in the CONTROL-M Active Jobs file, the IOA Conditions file, and the CONTROL-M Resources file that occur during the CONTROL-M working day.

The Journal file is initialized each day during New Day processing. From that point on, for the rest of the working day, the CONTROL-M monitor records in the Journal file all job processing activities that impact the CONTROL-M Active Jobs file, and all prerequisite condition additions to and deletions from the IOA Conditions file and the CONTROL-M Resources file.

If the CONTROL-M Active Jobs file, and optionally, the IOA Conditions file and the CONTROL-M Resources file, need to be restored, for example, following a system crash, utility CTMRSTR can be run to restore the files. The utility uses data from the Journal file to restore the files to the status they had at any specific time after the last run of the New Day procedure.

The CONTROL-M Journal file is initialized each day during New Day processing. Therefore, the time at which the New Day procedure initialized the Journal file is the earliest time to which the CONTROL-M Active Jobs file, the CONTROL-M Resources file, or the IOA Conditions file can be restored.

Journaling and Restoration is an optional feature that can be activated by the INCONTROL administrator. It is described in the CONTROL-M chapter of the *INCONTROL for z/OS Administrator Guide*, and activation of this feature is described under parameter JRNL in the chapter on installing CONTROL-M in the *INCONTROL for z/OS Installation Guide*.

Accumulating Statistics: Statistics Facility

As part of the postprocessing for each job, CONTROL-M determines the elapsed run time of the job. All accumulated information regarding job execution, including the elapsed run time, is written to the IOA Log file.

Periodically, statistics utility CTMJSA, which was mentioned in [Chapter 6, “Navigating The Active Environment,”](#) and which is described in detail in the CONTROL-M chapter of the *INCONTROL for z/OS Utilities Guide*, can be used to scan and analyze the IOA Log file. This utility gathers information about start time, elapsed run time, CPU utilization time, and so on, for each job. The utility places this information in the Statistics file, where averages of these values can be maintained for each job.

Statistics facility averages can be used for several purposes, such as

- determining if the execution time of a job falls outside a statistically normal range of time, which would indicate an execution delay or problem
- determining when a shout message should be issued based on the elapsed time of a job
- simulating job executions and forecast the impact of changes to the system.

- determining if a job can complete execution before the CONTROL-M planned shutdown time (QUIESCE command)
- calculating DUE-IN time for use by the Deadline Scheduling facility

For more information, see the discussion on Automatic Job Flow Adjustment in the introductory chapter of the *CONTROL-M for z/OS User Guide*.

Automatic Tape Adjustment

The Automatic Tape Adjustment facility collects and analyzes statistics regarding tape drive usage, and automatically allocates the appropriate number of tape drives at job order time. This facility, which can be implemented by your INCONTROL administrator, overrides any tape drive Quantitative resource value specified in the job scheduling definition. For more information, see the discussion about tape device usage statistics in the description of the Statistics screen and the description of the RESOURCE parameter in the *CONTROL-M for z/OS User Guide*.

Simulating Job Execution and Forecasting Resource Usage

Using statistics accumulated by the Statistics facility, the Simulation and Forecasting facility simulates the actions of the CONTROL-M monitor under the conditions specified in simulation parameters.

The Simulation and Forecasting facility enables you to forecast anticipated job load for a specified time in the future, and to forecast the effects that possible changes to the system might have.

The Simulation and Forecasting facility can improve the efficiency of your site. It can help with resource and configuration decisions, and it can help with the planning of workload scheduling to achieve maximum utilization of resources.

The Simulation and Forecasting facility is described in the chapter on the Simulation and Forecasting facility in the *CONTROL-M for z/OS User Guide*.

The facility utilizes the Simulation and Forecasting procedure that can be activated by online utility CTMCSIM, which is run by Option M3 in the IOA Utility menu. This utility, which can also activate the Tape Pull List procedure, is described in the IOA Utilities chapter of the *CONTROL-M for z/OS User Guide*.

CONTROL-M/Restart Dataset Cleanup Utility CTCCLN (R2)

Utility CTCCLN, the CONTROL-M/Restart Dataset Cleanup Utility, is used to manually request dataset cleanup. The utility places a CONTROLR step in the job stream and submits the job. The CONTROLR step performs the dataset adjustment, including step adjustment, if necessary, and then stops. No further job steps are executed. The utility adds a step that prevents the execution of the other steps in the job.

This utility can be requested by entering option R2 in the IOA Utility menu. The utility is described in detail in the CONTROLR step chapter of the *CONTROL-M/Restart User Guide*.

Reporting Facility

CONTROL-M supports a comprehensive reporting facility, which can produce the following types of reports:

Table 29 Report Types

Reports	Description
Keystroke Language Reports	These are reports generated with the Keystroke Language (KSL). KSL is a general purpose reporting language, based on the Online facility, capable of producing numerous reports from the database.
Special Purpose Reports	These reports include the Job Flow reports that are generally used to track the dependencies between jobs, and the Job Plan reports that are used to anticipate which jobs are scheduled each day.

Sample reports are provided in the IOA SAMPLE library. The Reporting facility is described in the Keystroke Language (KSL) chapter of the *CONTROL-M for z/OS User Guide*. Special purpose reports are described in the *INCONTROL for z/OS Utilities Guide*.

Exit the Online Session

39 Exit the Online facility.

This completes the steps in this chapter of the *CONTROL-M for z/OS Getting Started Guide*.

Review

In this chapter you

- created a calendar in the IOA Calendar facility and used it to schedule a job
- created a CMEM table and rule, and loaded the rule to memory
- used a CMEM rule to bring a job On Spool in CMEM
- used batch utility IOACND to cleanup conditions and resources
- learned about the following facilities and capabilities
 - History Jobs File
 - Journaling and Restoration Capability
 - Accumulating Statistics: Statistics Facility
 - Automatic Tape Adjustment
 - Simulating Job Execution and Forecasting Resource Usage and utility CTMCSIM
 - CONTROL-M/Restart Dataset cleanup utility CTRCCLN
 - Reporting Facility

Recommended Reading

It is recommended that you read the following:

- In the *CONTROL-M for z/OS User Guide*:
 - In Chapter 2, the description of:
 - IOA Calendar facility
 - CMEM facility
 - CONTROL-M History Jobs file
 - CONTROL-M Statistics file (and screen)
 - Online utility CTMCSIM (M3)
 - In Chapter 4, the description of the CMEM facility, especially relating to On Spool Jobs
 - In Chapter 7, CONTROL-M Simulation and Forecast Facility
- In the *CONTROL-M/Restart User Guide*:
 - In Chapter 2, the description of online utility CTRCCLN (R2)
- In the *CONTROL-M for z/OS Utilities Guide*,
 - In Chapter 2, the description of utility IOACND
 - In Chapter 3, the description of utilities CTMJSA

Sample JCLs

Shown below are sample JCLs used for the exercises in this book, according to chapter and job. The samples indicate how the JCL appears at the time of preparation. They do not indicate changes made during the course of the exercises. It may be necessary to customize some of these JCLs, particularly when names, such as a site-defined library name, must be specified.

Chapter 1

JOB1

```

EDIT          CTMP. JCL(M21JOB1) - 01.05          Columns 00001 00072
*****      ***** Top of Data *****
000001 //M21  JOB  APERLMAN, CLASS=A,
000002 //      MSGCLASS=X, NOTIFY=M21
000003 //S1  EXEC  PGM=I EFBR14
*****      ***** Bottom of Data *****

```

JOB2

```

EDIT          CTMP. JCL(M21JOB2) - 01.00          Columns 00001 00072
*****      ***** Top of Data *****
000001 //M21  JOB  APERLMAN, CLASS=A,
000002 //      MSGCLASS=X, NOTIFY=M21
000003 //S1  EXEC  PGM=I EFBR14
*****      ***** Bottom of Data *****

```

Chapter 2

JOB3

```

EDIT          CTMP.JCL(M21JOB3) - 01.03          Columns 00001 00072
*****      ***** Top of Data *****
000100 //M21 JOB , ' A. PERLMAN' , CLASS=A, MSGCLASS=X, NOTI FY=M21
001400 //S1 EXEC PGM=I OATEST, PARM=' TERM=C0008'
001500 //STEPLI B DD DSN=I OAP. LOAD, DI SP=SHR
001600 // DD DI SP=SHR, DSN=SCM. DEV. I 600. LOAD
001700 //DAPARM DD DI SP=SHR, DSN=I OAP. PARM
001800 // DD DI SP=SHR, DSN=SCM. DEV. I 600. I OA. I OAEV
001900 //DALOG DD DI SP=SHR, DSN=I OAP. LOG
002000 //SYSPRI NT DD SYSOUT=*
*****      ***** Bottom of Data *****

```

JOB4

```

EDIT          CTMP.JCL(M21JOB4) - 01.00          Columns 00001 00072
*****      ***** Top of Data *****
000001 //M21 JOB APERLMAN, CLASS=A,
000002 // MSGCLASS=X, NOTI FY=M21
000003 //S1 EXEC PGM=I EFBR14
*****      ***** Bottom of Data *****

```

Chapter 3

JOB5

```

EDIT          CTMP.JCL(M21JOB5) - 01.09                      Columns 00001 00072
*****      ***** Top of Data *****
000001 //M21 JOB , ' A. PERLMAN' , CLASS=A, MSGCLASS=X, NOTI FY=M21
000002 //S1 EXEC PGM=I OATEST, PARM=' TERM=C0000'
000003 //STEPLI B DD DSN=I OAP. LOAD, DI SP=SHR
000004 // DD DI SP=SHR, DSN=SCM. DEV. I 600. LOAD
000005 //DAPARM DD DI SP=SHR, DSN=I OAP. PARM
000006 // DD DI SP=SHR, DSN=SCM. DEV. I 600. I OA. I OAENV
000007 //DALOG DD DI SP=SHR, DSN=I OAP. LOG
000008 //SYSPRI NT DD SYSOUT=*
000009 //S2 EXEC PGM=I OATEST, PARM=' TERM=C0000'
000010 //STEPLI B DD DSN=I OAP. LOAD, DI SP=SHR
000011 // DD DI SP=SHR, DSN=SCM. DEV. I 600. LOAD
000012 //DAPARM DD DI SP=SHR, DSN=I OAP. PARM
000013 // DD DI SP=SHR, DSN=SCM. DEV. I 600. I OA. I OAENV
000014 //DALOG DD DI SP=SHR, DSN=I OAP. LOG
000015 //SYSPRI NT DD SYSOUT=*
000016 //S3 EXEC PGM=I OATEST, PARM=' TERM=C0008'
000017 //STEPLI B DD DSN=I OAP. LOAD, DI SP=SHR
000018 // DD DI SP=SHR, DSN=SCM. DEV. I 600. LOAD
000019 //DAPARM DD DI SP=SHR, DSN=I OAP. PARM
000020 // DD DI SP=SHR, DSN=SCM. DEV. I 600. I OA. I OAENV
000021 //DALOG DD DI SP=SHR, DSN=I OAP. LOG
000022 //SYSPRI NT DD SYSOUT=*
000023 //S4 EXEC PGM=I OATEST, PARM=' TERM=C0000'
000024 //STEPLI B DD DSN=I OAP. LOAD, DI SP=SHR
000025 // DD DI SP=SHR, DSN=SCM. DEV. I 600. LOAD
000026 //DAPARM DD DI SP=SHR, DSN=I OAP. PARM
000027 // DD DI SP=SHR, DSN=SCM. DEV. I 600. I OA. I OAENV
000028 //DALOG DD DI SP=SHR, DSN=I OAP. LOG
000029 //SYSPRI NT DD SYSOUT=*
000030 //S5 EXEC PGM=I OATEST, PARM=' TERM=C0000'
000031 //STEPLI B DD DSN=I OAP. LOAD, DI SP=SHR
000032 // DD DI SP=SHR, DSN=SCM. DEV. I 600. LOAD
000033 //DAPARM DD DI SP=SHR, DSN=I OAP. PARM
000034 // DD DI SP=SHR, DSN=SCM. DEV. I 600. I OA. I OAENV
000035 //DALOG DD DI SP=SHR, DSN=I OAP. LOG
000036 //SYSPRI NT DD SYSOUT=*
*****      ***** Bottom of Data *****

```

Chapter 4

JOB1

```

EDIT          CTMP. JCL(M21JOB1) - 01.05          Columns 00001 00072
*****      ***** Top of Data *****
000001 //M21  JOB APERLMAN, CLASS=A,
000002 //      MSGCLASS=X, NOTIFY=M21
000003 //S1  EXEC  PGM=IEFBR14
*****      ***** Bottom of Data *****

```

JOB2

```

EDIT          CTMP. JCL(M21JOB2) - 01.00          Columns 00001 00072
*****      ***** Top of Data *****
000001 //M21  JOB APERLMAN, CLASS=A,
000002 //      MSGCLASS=X, NOTIFY=M21
000003 //S1  EXEC  PGM=IEFBR14
*****      ***** Bottom of Data *****

```

JOB3

```

EDIT          CTMP. JCL(M21JOB3) - 01.03          Columns 00001 00072
*****      ***** Top of Data *****
000100 //M21  JOB , ' A. PERLMAN' , CLASS=A, MSGCLASS=X, NOTIFY=M21
001400 //S1  EXEC  PGM=IOATEST, PARM=' TERM=C0000'
001500 //STEPLIB DD   DSN=IOAP. LOAD, DISP=SHR
001600 //      DD   DISP=SHR, DSN=SCM. DEV. I600. LOAD
001700 //DAPARM DD   DISP=SHR, DSN=IOAP. PARM
001800 //      DD   DISP=SHR, DSN=SCM. DEV. I600. IOA. IOAENV
001900 //DALOG  DD   DISP=SHR, DSN=IOAP. LOG
002000 //SYSPRINT DD   SYSOUT=*
*****      ***** Bottom of Data *****

```

JOB4

```
EDIT          CTMP. JCL(M21JOB4) - 01.00          Columns 00001 00072
*****      ***** Top of Data *****
000001 //M21  JOB  APERLMAN, CLASS=A,
000002 //      MSGCLASS=X, NOTIFY=M21
000003 //S1  EXEC  PGM=IEFBR14
*****      ***** Bottom of Data *****
```

JOB5

```

EDIT          CTMP: JCL(M21JOB5) - 01.09                      Columns 00001 00072
*****      ***** Top of Data *****
000001 //M21 JOB , ' A. PERLMAN' , CLASS=A, MSGCLASS=X, NOTIFY=M21
000002 //S1 EXEC PGM=I OATEST, PARM=' TERM=C0000'
000003 //STEPLI B DD DSN=I OAP. LOAD, DI SP=SHR
000004 // DD DI SP=SHR, DSN=SCM. DEV. I 600. LOAD
000005 //DAPARM DD DI SP=SHR, DSN=I OAP. PARM
000006 // DD DI SP=SHR, DSN=SCM. DEV. I 600. I OA. I OAENV
000007 //DALOG DD DI SP=SHR, DSN=I OAP. LOG
000008 //SYSPRI NT DD SYSOUT=*
000009 //S2 EXEC PGM=I OATEST, PARM=' TERM=C0000'
000010 //STEPLI B DD DSN=I OAP. LOAD, DI SP=SHR
000011 // DD DI SP=SHR, DSN=SCM. DEV. I 600. LOAD
000012 //DAPARM DD DI SP=SHR, DSN=I OAP. PARM
000013 // DD DI SP=SHR, DSN=SCM. DEV. I 600. I OA. I OAENV
000014 //DALOG DD DI SP=SHR, DSN=I OAP. LOG
000015 //SYSPRI NT DD SYSOUT=*
000016 //S3 EXEC PGM=I OATEST, PARM=' TERM=C0008'
000017 //STEPLI B DD DSN=I OAP. LOAD, DI SP=SHR
000018 // DD DI SP=SHR, DSN=SCM. DEV. I 600. LOAD
000019 //DAPARM DD DI SP=SHR, DSN=I OAP. PARM
000020 // DD DI SP=SHR, DSN=SCM. DEV. I 600. I OA. I OAENV
000021 //DALOG DD DI SP=SHR, DSN=I OAP. LOG
000022 //SYSPRI NT DD SYSOUT=*
000023 //S4 EXEC PGM=I OATEST, PARM=' TERM=C0000'
000024 //STEPLI B DD DSN=I OAP. LOAD, DI SP=SHR
000025 // DD DI SP=SHR, DSN=SCM. DEV. I 600. LOAD
000026 //DAPARM DD DI SP=SHR, DSN=I OAP. PARM
000027 // DD DI SP=SHR, DSN=SCM. DEV. I 600. I OA. I OAENV
000028 //DALOG DD DI SP=SHR, DSN=I OAP. LOG
000029 //SYSPRI NT DD SYSOUT=*
000030 //S5 EXEC PGM=I OATEST, PARM=' TERM=C0000'
000031 //STEPLI B DD DSN=I OAP. LOAD, DI SP=SHR
000032 // DD DI SP=SHR, DSN=SCM. DEV. I 600. LOAD
000033 //DAPARM DD DI SP=SHR, DSN=I OAP. PARM
000034 // DD DI SP=SHR, DSN=SCM. DEV. I 600. I OA. I OAENV
000035 //DALOG DD DI SP=SHR, DSN=I OAP. LOG
000036 //SYSPRI NT DD SYSOUT=*
*****      ***** Bottom of Data *****

```

Chapter 5

JOB6

```

EDIT          CTMP. JCL(M21JOB6) - 01.05          Columns 00001 00072
*****      ***** Top of Data *****
000001 //M21  JOB APERLMAN, CLASS=A,
000002 //      MSGCLASS=X, NOTI FY=M21
000003 //S1  EXEC  PGM=I EFBR14
*****      ***** Bottom of Data *****

```

JOB7

```

EDIT          CTMP. JCL(M21JOB7) - 01.05          Columns 00001 00072
*****      ***** Top of Data *****
000001 //M21  JOB APERLMAN, CLASS=A,
000002 //      MSGCLASS=X, NOTI FY=M21
000003 //S1  EXEC  PGM=I EFBR14
*****      ***** Bottom of Data *****

```

JOB8

```

EDIT          CTMP. JCL(M21JOB8) - 01.05          Columns 00001 00072
*****      ***** Top of Data *****
000001 //M21  JOB APERLMAN, CLASS=A,
000002 //      MSGCLASS=X, NOTI FY=M21
000003 //S1  EXEC  PGM=I EFBR14
*****      ***** Bottom of Data *****

```

JOB9

```

EDIT          CTMP. JCL(M21JOB9) - 01.05          Columns 00001 00072
*****      ***** Top of Data *****
000001 //M21  JOB APERLMAN, CLASS=A,
000002 //      MSGCLASS=X, NOTI FY=M21
000003 //S1  EXEC  PGM=I EFBR14
*****      ***** Bottom of Data *****

```

Chapter 6

JOB3

```
EDIT          CTMP.JCL(M21JOB3) - 01.03          Columns 00001 00072
*****      ***** Top of Data *****
000100 //M21 JOB , ' A. PERLMAN' , CLASS=A, MSGCLASS=X, NOTIFY=M21
001400 //S1 EXEC PGM=I OATEST, PARM=' TERM=C0008'
001500 //STEPLIB DD DSN=I OAP. LOAD, DISP=SHR
001600 // DD DISP=SHR, DSN=SCM. DEV. I 600. LOAD
001700 //DAPARM DD DISP=SHR, DSN=I OAP. PARM
001800 // DD DISP=SHR, DSN=SCM. DEV. I 600. I OA. I OAENV
001900 //DALOG DD DISP=SHR, DSN=I OAP. LOG
002000 //SYSPRINT DD SYSOUT=*
*****      ***** Bottom of Data *****
```

JOB5

```

EDIT          CTMP. JCL(M21JOB5) - 01.09                      Columns 00001 00072
*****
***** Top of Data *****
000001 //M21 JOB , ' A. PERLMAN' , CLASS=A, MSGCLASS=X, NOTIFY=M21
000002 //S1 EXEC PGM=I OATEST, PARM=' TERM=C0000'
000003 //STEPLI B DD DSN=I OAP. LOAD, DI SP=SHR
000004 // DD DI SP=SHR, DSN=SCM. DEV. I 600. LOAD
000005 //DAPARM DD DI SP=SHR, DSN=I OAP. PARM
000006 // DD DI SP=SHR, DSN=SCM. DEV. I 600. I OA. I OAENV
000007 //DALOG DD DI SP=SHR, DSN=I OAP. LOG
000008 //SYSPRI NT DD SYSOUT=*
000009 //S2 EXEC PGM=I OATEST, PARM=' TERM=C0000'
000010 //STEPLI B DD DSN=I OAP. LOAD, DI SP=SHR
000011 // DD DI SP=SHR, DSN=SCM. DEV. I 600. LOAD
000012 //DAPARM DD DI SP=SHR, DSN=I OAP. PARM
000013 // DD DI SP=SHR, DSN=SCM. DEV. I 600. I OA. I OAENV
000014 //DALOG DD DI SP=SHR, DSN=I OAP. LOG
000015 //SYSPRI NT DD SYSOUT=*
000016 //S3 EXEC PGM=I OATEST, PARM=' TERM=C0000'
000017 //STEPLI B DD DSN=I OAP. LOAD, DI SP=SHR
000018 // DD DI SP=SHR, DSN=SCM. DEV. I 600. LOAD
000019 //DAPARM DD DI SP=SHR, DSN=I OAP. PARM
000020 // DD DI SP=SHR, DSN=SCM. DEV. I 600. I OA. I OAENV
000021 //DALOG DD DI SP=SHR, DSN=I OAP. LOG
000022 //SYSPRI NT DD SYSOUT=*
000023 //S4 EXEC PGM=I OATEST, PARM=' TERM=C0000'
000024 //STEPLI B DD DSN=I OAP. LOAD, DI SP=SHR
000025 // DD DI SP=SHR, DSN=SCM. DEV. I 600. LOAD
000026 //DAPARM DD DI SP=SHR, DSN=I OAP. PARM
000027 // DD DI SP=SHR, DSN=SCM. DEV. I 600. I OA. I OAENV
000028 //DALOG DD DI SP=SHR, DSN=I OAP. LOG
000029 //SYSPRI NT DD SYSOUT=*
000030 //S5 EXEC PGM=I OATEST, PARM=' TERM=C0000'
000031 //STEPLI B DD DSN=I OAP. LOAD, DI SP=SHR
000032 // DD DI SP=SHR, DSN=SCM. DEV. I 600. LOAD
000033 //DAPARM DD DI SP=SHR, DSN=I OAP. PARM
000034 // DD DI SP=SHR, DSN=SCM. DEV. I 600. I OA. I OAENV
000035 //DALOG DD DI SP=SHR, DSN=I OAP. LOG
000036 //SYSPRI NT DD SYSOUT=*
*****
***** Bottom of Data *****

```

Chapter 7

No special preparations required.

Chapter 8

JOB10

```

EDIT          CTMP. JCL(M21JOB10) - 01.00                      Columns 00001 00072
*****      ***** Top of Data *****
000001 //M21  JOB  APERLMAN, CLASS=A,
000002 //      MSGCLASS=X, NOTI FY=M21
000003 //S1   EXEC  PGM=I EFBR14
*****      ***** Bottom of Data *****

```

EXT1

```

EDIT          CTMP. JCL(M21EXT1) - 01.09                      Columns 00001 00072
*****      ***** Top of Data *****
000001 //M21  JOB  , ' A. PERLMAN', CLASS=A, MSGCLASS=X, NOTI FY=M21
000002 //S1   EXEC  PGM=I OATEST, PARM=' TERM=C0000'
000003 //STEPLI B DD   DSN=I OAP. LOAD, DI SP=SHR
000004 //          DD   DI SP=SHR, DSN=SCM. DEV. I 600. LOAD
000005 //DAPARM DD   DI SP=SHR, DSN=I OAP. PARM
000006 //          DD   DI SP=SHR, DSN=SCM. DEV. I 600. I OA. I OAENV
000007 //DALOG DD   DI SP=SHR, DSN=I OAP. LOG
000008 //SYSPRI NT DD   SYSOUT=*
000009 //S2   EXEC  PGM=I OATEST, PARM=' TERM=C0000'
000010 //STEPLI B DD   DSN=I OAP. LOAD, DI SP=SHR
000011 //          DD   DI SP=SHR, DSN=SCM. DEV. I 600. LOAD
000012 //DAPARM DD   DI SP=SHR, DSN=I OAP. PARM
000013 //          DD   DI SP=SHR, DSN=SCM. DEV. I 600. I OA. I OAENV
000014 //DALOG DD   DI SP=SHR, DSN=I OAP. LOG
000015 //SYSPRI NT DD   SYSOUT=*
*****      ***** Bottom of Data *****

```

Index

Symbols

%% SET control statement [141](#)
 %%\$CALCDTE AutoEdit function [141](#), [142](#)
 %%\$CALCDTE function [141](#), [142](#)
 %%\$JULIAN AutoEdit function [142](#)
 %%\$JULIAN function [142](#)
 %%A AutoEdit variable [141](#)

A

Active Environment screen [36](#)
 Changing display types [175](#)
 Checking the job log [41](#)
 Holding and deleting job orders [44](#)
 ADD command [82](#)
 AutoEdit
 Checking syntax and results [146](#)
 Complex terms [155](#)
 Control statements [154](#)
 Control statements and functions [140](#)
 Functions [142](#), [144](#)
 %%\$CALCDTE [141](#), [142](#)
 %%\$JULIAN [142](#)
 Operators [156](#)
 Using system variables in the JCL [135](#)
 Variables
 %%A [141](#)
 Job scheduling definition statements [150](#)
 Non-Date system [139](#)
 Automatic Tape Adjustment facility [234](#)

B

Basis scheduling parameters [60](#)
 BMC Software, contacting [2](#)

C

Calendars, defining [209](#)
 CMEM rule, creating [224](#)
 Commands
 ADD [82](#)
 DOWN [73](#)
 NOTE [176](#)

 UP [73](#)
 Conditions [70](#)
 adding [82](#)
 Control statements
 %% SET [141](#)
 AutoEdit [154](#)
 CONTROL-M
 Active Jobs file [233](#)
 Journal file [232](#)
 Main components [24](#)
 Reporting facility [235](#)
 Restoring files [233](#)
 CONTROL-M/Restart Dataset Cleanup Utility CTCRCLN (R2) [235](#)
 Conventions Used in This Guide [17](#)
 Creating
 CMEM Rule [224](#)
 Job JCL [224](#)
 Job scheduling definition [226](#)
 On Spool jobs [222](#)
 CTMAESIM utility [146](#)
 CTMJOB utility [194](#)
 CTMJOBREQ utility [203](#)
 CTMRSTR utility [233](#)
 CTCRCLN (R2) Dataset Cleanup utility [235](#)
 customer support [3](#)

D

Date qualifiers [70](#)
 Date system variables [138](#)
 Defining
 Calendars [209](#)
 Date Control Record for a User Daily job [199](#)
 Global variables [158](#)
 JCL of a User Daily job [198](#)
 Job scheduling definition of a User Daily job [201](#)
 JOB4 [74](#)
 On Spool jobs [223](#)
 Restart in the job scheduling definition [93](#)
 Displaying
 Execution information from job runs [182](#)
 Job dependencies [180](#)
 Jobs belonging to a specific group [189](#)
 List of Available Filters [173](#)

Scheduling plan for the Job 71
 Statistics for a job 188
 Table names for jobs 179
 DO statement, CMEM rule 221

E

Enhanced Daily Checkpointing 200
 Exiting the Online facility 50
 External members, pointing to 153

F

Facilities, miscellaneous 232
 Filtering
 Active Environment display 37
 IOA Conditions/Resources screen 81
 Filters 169
 Displaying list of available 173
 Forecasting resource usage 234
 Functions 144
 %%SCALCDTE 141, 142
 %%SJULIAN 142

G

General parameters 59
 Global variables
 Access by another job 160
 Defining 158
 Syntax 159
 Update by another job 162
 Group
 Entity 108
 Ordering 126
 Logic of the Entity scheduling definition 114
 Scheduling
 Advantages 130
 Creating table 107
 Points to remember 126

H

History Jobs File 232

I

IOA Calendar Definition facility 214
 IOA Conditions/Resources screen 81
 IOACND batch utility 229

J

JCL
 Editing from the Active Environment 99
 Setup operation flow 163
 Job List Screen, formats 50
 Job Order Execution History screen 182
 Job Ordering methods 205
 Job scheduling definition facility, returning to 46
 Job scheduling definitions
 Defining restart in 93
 Facility 26
 In the Group table 115
 Planned logic of 115
 Saving 46
 Supplying variable values 144
 JOB4, defining 74
 Jobs
 Checking the Sysout for 42
 Confirming an order 39
 Creating JCL 224
 Creating scheduling definition 226
 Dependencies 70
 Displaying
 Belonging to a specific group 189
 Dependencies 180
 Execution information from runs 182
 Scheduling plan 71
 Statistics for 188
 Table names 179
 Forcing 76
 Forcing an OK status for 186
 Freeing a held job 86
 Identifying scheduling problems 79
 On Spool 228
 Bringing 227
 Creating 222
 Defining 223
 Ordering 76, 126
 Through online utility CTMJOBQRQ 203
 Through utility CTMJOB 194
 Restarting 100
 Simulating execution 234
 User Daily 198, 201
 Defining Date Control Record 199
 Defining JCL 198
 Defining job scheduling definition 201
 Journal file, CONTROL-M 232

N

New Day processing 197
 NOTE command 176
 Note, adding to a job 176

O

ODATE 137
 ON GROUP-END parameter 112
 On Spool jobs 228
 ON Statements 221
 Online facility
 Exiting 50
 Primary options 26

P

Parameters
 Basis scheduling 60
 General 59
 ON GROUP-END 112
 Post-processing 112
 Runtime scheduling
 section 64
 Post-processing
 Parameters 112
 Parameters section 67
 product support 3

R

Reporting facility, CONTROL_M 235
 Resource usage, forecasting 234
 Resource, adding 82
 Responding to external events through CMEM 215
 Runtime Scheduling Parameters 111

S

Saving the Table 126
 Schedule Tags and Other Basic Scheduling Parameters 109
 Screens
 Active Environment 36
 Changing display types 175
 Checking the job log 41
 Holding and deleting job orders 44
 IOA Conditions/Resources 81
 Filtering 81
 Job List, formats 50
 Job Order Execution History 182
 Sysout Viewing 184
 Table List, selecting a table from 48
 Why 78, 83
 Zoom 84
 Statements, ON 221
 Statistics, accumulating 233
 support, customer 3
 Sysout Viewing screen 184
 System date 137

T

Table List screen
 Selecting a table from 48
 technical support 3
 Toggling Between Display of Commands and Options 36

U

User Daily jobs 198, 201
 Utilities
 CTMAESIM 146
 CTMJOB 194
 CTMJOBREQ 203
 CTMRSTR 233
 CTRCLN (R2) 235
 IOACND 229
 Utility CTMAESIM 146

V

Variables
 %%A 141
 Date 138
 Global
 Access by another job 160
 Defining 158
 Syntax 159
 Update by another job 162
 Local and global 156
 Non-Date system 139
 Supplying values through an external member 151
 Supplying values through the job scheduling
 definition 144

W

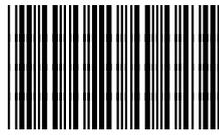
Why screen 78, 83
 Adding a missing condition in 80
 Working date 137

Z

Zoom screen 84

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Notes



62362