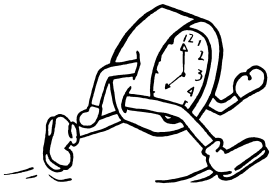
PJS

Personal Job Scheduler



User's Guide

Release 3.1

Draft 2/9/05

Personal Job Scheduler (PJS) 3.1.0 was released for distribution in February, 2005.

MVS is a trademark and IBM®, OS/390®, RACF®, and z/OS® are registered trademarks of International Business Machines Corporation.

Copyright © Northrop Grumman, 1990, 2005. All rights reserved.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with the Invariant Sections being "PJS Software Copyright and License" and "GNU General Public License", with no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Table of Contents

About This Manual ..... **ix**
Acknowledgments ix
PJS Software Copyright and License ix
Comments and Suggestions ix
Organization of This Manual x
Related Publications xi
Notational Symbols xii

Chapter 1: PJS Overview **1**
PJS Terms and Concepts 1
Job Request 1
Owner 2
PJS Request Queue 2
PJS System Task 2

Chapter 2: Using PJS to Schedule Jobs **3**
Creating a Job Request 3
The Job Request-ID 3
The Job Request Description 3
The Job Request Status 4
Specifying the JCL Source 4
The JCL Data Set 4
Using the PJS JCL Spool 5
Specifying When to Submit the Job 5
Specifying the Start Date and Time 5
Specifying the End Date and Time 5
Specifying the Frequency 6
Specifying a Submit Window 8
Using Events to Control Job Submission 9
Creating an Event 9
Job Request Events 9
Preposted and Non-Preposted Events 10
Posting an Event 10
Resetting an Event 11
Event Status 11
Modifying the JCL with Variable Substitution 12
Specifying the Target String in the Source JCL 12
Specifying the Replacement Value 13
Specifying the Data Shift Options 14
Controlling PJS Messages 17
Sending Message to a TSO User 17
Filtering Messages by Severity Level 17
Using the PJS Message History Log 18

Chapter 3: PJS Processing Overview **19**
Updating the PJS Database 20
Submitting a Batch Job 21

Posting an Event....22

Chapter 4: PJS Specification Conventions....23
Record-ID Specifications....23
The Job Request-ID....23
The Calendar-ID....24
The Event-ID....24
The Global Variable-ID....25
Date Specifications....25
Absolute Dates....25
Relative Dates....26
Time Specifications....26
Absolute Times....26
Relative Times....27

Chapter 5: The PJS TSO Commands....29
PJS Job Request Commands....29
PJREQADD....30
PJREQDEL....31
PJREQHST....32
PJREQLST....33
PJREQMOD....34
PJREQADD and PJREQMOD Subcommands....35
CANCEL Subcommand....36
END Subcommand....37
EVENT Subcommand....38
LIST Subcommand....40
RESET Subcommand....41
SET Subcommand....42
VARIABLE Subcommand....47
PJS Calendar Commands....50
PJCALADD....51
PJCALDEL....52
PJCALIST....53
PJCALMOD....54
PJS Event Commands....56
PJEVLIST....57
PJEVPOST....58
PJEVRSET....59
PJS Global Variable Commands....60
PJVARADD....61
PJVARDEL....62
PJVARLST....63
PJVARMOD....64

Chapter 6: The PJS ISPF Interface....65
General Panels....66
The PJS Main Menu Panel....66
PJS Tutorial and Help Panels....68
Job Request Panels....69
Job Request Menu Panel....69
List Job Requests for Owner Panel....71

Add Job Request Panel.... 75
Add Job Request-ID Panel....79
Modify Job Request Panel.... 80
Delete Job Request Panel....83
Display Job Request Panel....85
Specify Job Request JCL Source Panel.... 87
Specify Job Request Frequency Panel.... 89
Specify Job Request Submit Options Panel....92
Specify Job Request Events Panel....95
Specify Job Request Variables Panel.... 98
Display Job Request Events Panel....102
Display Job Request Variables Panel.... 104
Display Job Request History Panel....107
Display Job Request Internal Information Panel.... 109
Calendar Panels....112
Calendar Menu Panel....112
List Calendars for Owner Panel....114
Add Calendar Panel.... 116
Add Calendar-ID Panel....118
Modify Calendar Panel.... 119
Delete Calendar Panel....122
Display Calendar Panel....124
List Job Requests for Calendar Panel.... 126
Event Panels.... 129
Event Menu Panel.... 129
List Events for Owner Panel.... 131
Post Event Panel.... 133
Reset Event Panel.... 135
Display Event Panel....137
List Job Requests for Event Panel.... 139
Global Variable Panels.... 142
Global Variable Menu Panel.... 142
List Global Variables Panel.... 144
Add Global Variable Panel.... 147
Add Variable-ID Panel.... 149
Modify Global Variable Panel....150
Delete Global Variable Panel.... 152
Display Global Variable Panel.... 154
List Job Requests for Global Variable Panel....156

Chapter 7: Examples....159
Example 1: How to Run a One-Time Job....161
Example 2: How to Run a Daily Job.... 171
Example 3: How to Run a Job on the First Tuesday of Each Month.... 181
Example 4: How to Run One Job After Another Job Completes....200
Example 5: How to Set Up Several dependent Jobs....221
Example 6: How to Run a Job After CICS Terminates....280
Example 7: How to Run a Weekly Job After a Daily Job....293
Example 8: How to Run a Job After a Manual Operation....314
Example 9: How to Enable a Failed Job Request....332
Example 10: How to Modify the Submitted JCL at Submit Time.... 348

Example 11: How to Use a Global Variable..... 361

Appendix A: Summary of Changes.....385
Changes for PJS Release 3.1.....385
Changes for PJS Release 2.1.4.....386
Changes for PJS Release 2.1.3.....386
Changes for PJS Release 2.1.....387

GNU General Public License..... 389

GNU Free Documentation License.....395

Figures

Figure 1: PJSPOST JCL....10
Figure 2: PJSRESET JCL.... 11
Figure 3: Variable Substitution Shifting Examples.... 16
Figure 4: Create Job Request Processing....20
Figure 5: Submitting a Batch Job....21
Figure 6: Posting an Event.... 22
Figure 7: PJREQADD Command Format....30
Figure 8: PJREQDEL Command Format....31
Figure 9: PJREQHST Command Format....32
Figure 10: PJREQLST Command Format.... 33
Figure 11: PJREQMOD Command Format.... 34
Figure 12: CANCEL Subcommand Format....36
Figure 13: END Subcommand Format....37
Figure 14: EVENT Subcommand Format....38
Figure 15: LIST Subcommand Format.... 40
Figure 16: RESET Subcommand Format....41
Figure 17: SET Subcommand Format....42
Figure 18: VARIABLE Subcommand Format....47
Figure 19: PJCALADD Command Format....51
Figure 20: PJCALDEL Command Format....52
Figure 21: PJCALIST Command Format.... 53
Figure 22: PJCALMOD Command Format.... 54
Figure 23: PJEVLIST Command Format....57
Figure 24: PJEVPOST Command Format.... 58
Figure 25: PJEVRSET Command Format.... 59
Figure 26: PJVARADD Command Format.... 61
Figure 27: PJVARDEL Command Format.... 62
Figure 28: PJVARLST Command Format....63
Figure 29: PJVARMOD Command Format....64
Figure 30: PJS Main Menu Panel.... 66
Figure 31: Job Request Menu Panel.... 69
Figure 32: List Job Requests for Owner Panel (Part 1 of 3)....71
Figure 33: List Job Requests for Owner Panel (Part 2 of 3)....72
Figure 34: List Job Requests for Owner Panel (Part 3 of 3)....72
Figure 35: Add Job Request Panel....76
Figure 36: Add Job Request-ID Panel.... 79
Figure 37: Modify Job Request Panel....80
Figure 38: Delete Job Request Panel.... 83
Figure 39: Display Job Request Panel.... 85
Figure 40: Specify Job Request JCL Source Panel....87
Figure 41: Specify Job Request Frequency Panel....89
Figure 42: Specify Job Request Submit Options Panel.... 92
Figure 43: Specify Job Request Events Panel....95
Figure 44: Specify Job Request Variables Panel....98
Figure 45: Display Job Request Events Panel.... 102
Figure 46: Display Job Request Variables Panel....104
Figure 47: Display Job Request History Panel.... 107

Figure 48: Display Job Request Internal Information Panel.....109
Figure 49: PJS Calendar Menu Panel..... 112
Figure 50: List Calendars for Owner Panel..... 114
Figure 51: Add Calendar Panel.....116
Figure 52: Add Calendar-ID Panel.....118
Figure 53: Modify Calendar Panel.....120
Figure 54: Delete Calendar Panel..... 122
Figure 55: Display Calendar Panel..... 124
Figure 56: Display Job Requests for Calendar Panel (Part 1 of 3).....126
Figure 57: Display Job Requests for Calendar Panel (Part 2 of 3).....127
Figure 58: Display Job Requests for Calendar Panel (Part 3 of 3).....127
Figure 59: PJS Event Menu Panel.....129
Figure 60: List Events for Owner Panel.....131
Figure 61: Post Event Panel..... 133
Figure 62: Reset Event Panel..... 135
Figure 63: Display Event Panel.....137
Figure 64: List Job Requests for Event Panel (Part 1 of 3)..... 139
Figure 65: List Job Requests for Event Panel (Part 2 of 3)..... 140
Figure 66: List Job Requests for Event Panel (Part 3 of 3)..... 140
Figure 67: PJS Global Variable Menu Panel..... 142
Figure 68: List Global Variables Panel (Part 1 of 2)..... 144
Figure 69: List Global Variables Panel (Part 2 of 2)..... 145
Figure 70: Add Global Variable Panel.....147
Figure 71: Add Variable ID Panel..... 149
Figure 72: Modify Global Variable Panel..... 150
Figure 73: Delete Global Variable Panel..... 152
Figure 74: Display Global Variable Panel..... 154
Figure 75: List Job Requests for Global Variable Panel (Part 1 of 3).....156
Figure 76: List Job Requests for Global Variable Panel (Part 2 of 3).....157
Figure 77: List Job Requests for Global Variable Panel (Part 3 of 3).....157

About This Manual

This manual describes how to use Personal Job Scheduler (PJS) Release 3.1. It is intended for anyone who wishes to use PJS to schedule jobs. It is assumed that the reader is familiar with OS/390 or z/OS batch jobs, including JCL, as well as how to use TSO or ISPF.

This manual also assumes that PJS has been installed with the default options, where applicable. It is possible that your installation has changed some of these options. There may also be security policies in place that affect how PJS works. See you site administrator for information about any local requirements.

Acknowledgments

PJS was written by Tim Henness

The original version of this manual was written by Matthew ???? and Tim Henness. Extensive revisions have been made by Tim Henness

PJS Software Copyright and License

All PJS software is **Copyright © Northrop Grumman, 1990, 2005. All rights reserved.**

PJS is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

PJS is distributed in the hope that it will be useful, but **WITHOUT ANY WARRANTY**; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

A copy of the GNU General Public License is included in the back of this book.

Comments and Suggestions

The author welcomes your comments and suggestions. He can be contacted at:

Tim Henness
Northrop Grumman IT
Internal Information Services
Bldg. 521-2
4101 Washington Ave.
Newport News, VA 23607

E-mail: Tim.Henness@ngc.com

Organization of This Manual

This manual contains the following chapters:

Chapter 1 "PJS Overview" introduces PJS and provides a bried overview of what PJS does, as well as some basic PJS terminology.

Chapter 2 "Using PJS to Schedule Jobs" provides a more detailed description of the capabilities provided by PJS. Explanations are given for most of the options that can be specified for PJS job requests, calendars, events, and variables.

Chapter 3 "PJS Processing Overview" describes how PJS does it's processing. Diagrams are included to help illustrate what happens when you use PJS.

Chapter 4 "PJS Specification Conventions" describes the syntax of some of the common parameters that are used in many of the PJS TSO comamnds and the ISPF dialogs, including how to enter record ID's and dates and times.

Chapter 5 "The PJS TSO Commands" describes the PJS TSO comamnds in detail.

Chapter 6 "The PJS ISPF Interface" describes the PJS ISPF panels in detail.

Chapter 7 "Examples" provides several examples of how to accmoplish some common tasks using PJS. Examples are provided for both TSO commands and the ISPF panels..

Appendix A "Summary of Changes" documents the changes made for each release of PJS that affect PJS users. Installation changes and problems fixed are listed in the ***PJS Installation Guide***.

Related Publications

PJS Documentation

Personal Job Scheduler (PJS) Installation Guide

Personal Job Scheduler (PJS) Messages and Codes

OS/390 and z/OS Documentation

MVS JCL User's Guide
OS/390 - GC28-1758, z/OS - SA22-7598

MVS JCL Reference
OS/390 - GC28-1757, z/OS - SA22-7597

TSO/E Command Reference
OS/390 - SC28-196, z/OS - SA22-7782

ISPF User's Guide Vol I
OS/390 - SC34-4791, z/OS - SC34-4822

ISPF User's Guide Vol II
OS/390 - SC34-4792, z/OS - SC34-4823

Notational Symbols

The following conventions are used in command formats throughout this manual:

BOLD UPPERCASE	is used to display commands or keywords you must code exactly as shown, for example, **SEND FILENAME.TXT**.
italic lowercase	is used to display information you must supply, for example, **SEND** *filename.txt*.
Underscores	either show a default value in a command description, display a default value in a screen image, or represent a highlighted word in a screen image.
Brackets [ ]	mean that you can select one of the items enclosed by the brackets; none of the enclosed items is required.
Braces { }	mean that you must select one of the items enclosed by the braces.
Vertical Bar |	separates options. One vertical bar separates two options, two vertical bars separate three options, and so on. You can select only one of the options.
Ellipsis . . .	means that you can repeat the word or clause that immediately precedes the ellipsis.

Chapter 1: PJS Overview

The Personal Job Scheduler (PJS) is a job scheduling system, that enables you to schedule your own batch jobs. PJS gives you complete control over the scheduling and submission of your batch jobs. PJS does not rely on a separate job scheduling and production control department

You can schedule a job to run on nearly any basis including:

- At any regular interval, from every minute, to every 99 years.
- At the end of each month (or any number of days before the last day of the month).
- On each of an arbitrary set of dates you specify.

Job submission can also be **event driven**, that is, the job is submitted after one or more external events (such as completion of another batch job) have completed.

PJS can modify the JCL as it is being submitted, replacing one character string in the JCL text with another, according to your specifications.

PJS Terms and Concepts

To understand PJS, you must understand some basic terms and concepts. This section is intended to introduce some of the more important terms and concepts. Only a brief description is given here. Each of these will be expanded in the subsequent chapters.

Job Request

If you want PJS to run a job for you, you must define a **Job Request**. A Job Request specifies such things as where PJS is to get the JCL to be submitted, when the job is to be submitted, if the job depends an any external events, and whether the JCL is to be modified when it is submitted.

Owner

PJS is designed to keep each user's Job Requests and other information separate from each other. By doing this, each user can identify their own Job Requests, Calendars, and other entities without concern for whether another user has an identically named entity. A PJS user is called an **Owner**.

PJS identifies each user with a unique **Owner-ID**. In most cases, the PJS Owner-ID is the same as the TSO User-ID. However, if a record is used by more than one TSO user, a site can allow a PJS Owner-ID to be a group-ID, a system-ID, or any other name assigned by the PJS Site Administrator.

An Owner-ID may be any name, 1 to 8 characters long. The first character must be alphabetic or a national character ('$', '#', or '@'). The remaining characters must be alphabetic, numeric, or a national character.

PJS Request Queue

PJS keeps most of its information in a system data set called the **PJS Request Queue**. When you issue a PJS TSO command or use the PJS ISPF interface, the TSO commands and ISPF dialogs access the PJS Request Queue. Information can be added, modified, deleted, or listed.

Most PJS information is kept in four PJS Request Queue record types:

- **Job Request Records** contain information about jobs and job submission.
- **Calendar Records** contain information about PJS calendars, which can be specified by job requests.
- **Event Records** contain information about the events specified in the Job Request Records.
- **Global Variable Records** contain information about the replacement values for modifying JCL at job submission.

Other record types are used for internal purposes. You need not be concerned with these other record types.

PJS System Task

The **PJS System Task** is a system component that normally runs whenever the system is running. The PJS System Task submits your jobs at the scheduled time. It also monitors the system for external events and performs other housekeeping tasks on the PJS databases.

Chapter 2: Using PJS to Schedule Jobs

This chapter is intended to give an overview of how you can use PJS to schedule jobs and the basic steps required accomplish this. This chapter does not provide detailed instructions of how to use the PJS TSO commands or the PJS ISPF panels. These are described in detail in **Chapter 5: The PJS TSO Commands** and **Chapter 6: The PJS ISPF Interface**.

Creating a Job Request

When you schedule a job to run with PJS, you must create a **Job Request**. Each job request in PJS describes a job to be submitted.

A Job Request is created with either the **PJREQADD** TSO command, or with the **Add Job Request** ISPF Panel. After a Job Request is created, it can be modified with the **PJREQMOD** TSO command or with the **Modify Job Request** ISPF Panel. It can be displayed by the **PJREQLST** TSO command or the **Display Job Request** ISPF panel. When it is no longer needed, it can be deleted with the **PJREQDEL** TSO command or the **Delete Job Request** ISPF panel.

The Job Request-ID

Each job request in PJS must have a unique **Job Request-ID** (or simply a **Request-ID**). The Request-ID consists of an **Owner-ID** and a **Request Name**. The Request Name may be a name chosen by the user, or, if one is not specified, PJS will assign a numeric name when the job request is created.

The Job Request-ID is usually formatted as *owner-ID***.***request-name*. A **Request Name** may be either a 1 to 8 character alphanumeric name, or a 3 digit number. If an alphanumeric name is specified, the first character must be alphabetic or a national character ('$', '#', or '@'). The remaining characters must be alphabetic, numeric, or a national character.

The Job Request Description

If desired, a job request can be given a 50-character description. The description can help you identify the purpose of a job request.

This description is used for documentation purposes only. It does not affect the operation of PJS in any way.

The Job Request Status

Each job request in PJS has a **Job Request Status** which indicates the current status of the job request.

The status can be one of the following:

WAIT indicates that the job request is waiting either for the next submit date and time, or for all of its job request events to be posted. this is the normal status for a job request.

COMPLETE indicates that PJS has completed all processing for the job request. The last date and time for submitting the job is past. The job request will remain in PJS until it is deleted.

DISABLED indicates that the job request has been disabled by the user. PJS will not process the job request until you enable it.

HOLD indicates that the job request has been held by the system. This can happen when PJS has been down for an extended period of time, or if a system problem required restoration of the PJS Request Queue, and the PJS Administrator felt that user action might be required to prevent duplicate jobs from being submitted. PJS will not process the job request until you enable it.

ERROR indicates the PJS encountered an error as it processed the job request. PJS will not process the job request until you enable it.

SUBMIT indicates that the PJS System Task is currently submitting the job for the job request. After job submission is complete, PJS will change the status to WAIT, COMPLETE, or ERROR.

The status is usually updated automatically by PJS as it processes the job request, but user actions can also affect the status. The PJS TSO commands or the PJS ISPF interface can be used to **disable** the job request to prevent it being processed until the user **enables** the job request (again using the PJS TSO commands or the PJS ISPF interface).

Specifying the JCL Source

The JCL Data Set

The **JCL Data Set** is the data set from which PJS obtains the JCL for the job to be submitted. The JCL data set may be a sequential data set, or a partitioned data set. If a partitioned data set is used, a member name must be specified.

PJS verifies that the JCL data set (and member, if a partitioned data set is specified) exists when the job request is created or modified. If it is not found, the add or modify will fail.

Using the PJS JCL Spool

When PJS submits a job, the JCL can be obtained in one of two ways. The JCL can be submitted directly from the user's JCL data set, or PJS can submit a copy of the JCL that was saved within PJS when the job request was added or modified.

First, PJS can read the JCL from the user's JCL data set when the job is submitted. If the JCL changes, the updated JCL will be obtained and submitted.

Second, PJS can read the JCL from the user's JCL data set when the job request is added or modified. A copy of the JCL is saved within a PJS facility called the **PJS JCL Spool**, and this copy is submitted. If the JCL in the user's JCL data set is changed, the save copy of the old JCL will continue to be submitted until the job request is modified specifying that the saved JCL is to be refreshed.

If the JCL is submitted directly from the user's JCL data set, and other users have update access to the data set, they could modify the JCL, which is submitted under your security User-ID, causing a security exposure. Saving the JCL in the PJS JCL Spool prevents this from happening.

Specifying When to Submit the Job

PJS determines when to submit a batch job by calculating the **Next Run Date and Time**. PJS calculates this value based on the **Start Date and Time**, the **End Date and Time**, the **Frequency**, and the date and time at which PJS performs the calculation. Each of these is covered in detail later.

Specifying the Start Date and Time

Each job request must have a **Start Date and Time**. This value defines the first date and time on which PJS is to submit the job.

If you specify a **Frequency** that conflicts with the **Start Date and Time**, or if the **Start Date and Time** is already past, PJS will submit the job on the next available date and time consistent with the **Frequency**.

Specifying the End Date and Time

If you want PJS to submit your batch job only until some future date and time, you can specify an **End Date and Time**. PJS will not schedule the job to run after the **End Date and Time**. However, the job can be submitted after the **End Date and Time** if the job was scheduled (the **Next Run Date and Time**) to run on or before the **End Date and Time**, but the submit was delayed (either by unposted events, or if

PJS is down at the scheduled date and time). If this value is not specified, the batch job will be submitted as specified by the **Frequency** until you change the Job Request (unless an error occurs).

Specifying the Frequency

The **Frequency** specifies how often PJS is to schedule the job. Several frequency options are available and are discussed below. A job request can specify only one frequency option.

One-Time Frequency

The simplest frequency option is called **Once**. This option schedules the job to run on time only, at the **Start Date and Time**. If and End Date and Time is specified, it is ignored. After the job is submitted, the job request is **COMPLETE**.

This can be useful if you wish to schedule a one-time job to perform a task at a time when you cannot be present to run the job yourself.

Periodic Frequency

A **Periodic** frequency is used to schedule a job to run at a regular interval. When you specify a **Periodic** frequency, you specify a **Quantity**, and a **Units**. The **Units** can be **Minutes**, **Hours**, **Days**, **Weeks**, **Months**, or **Years**. The **Quantity** specifies how may of each **Units** elapse between each job submission.

When a **Periodic** frequency is specified, the job is first scheduled at the **Start Date and Time**. The job is then rescheduled by adding the **Quantity** of **Units** specified to get the new **Next Run Date and Time**. The **Next Run Date and Time** will always be a multiple of the periodic **Quantity** and **Units**, plus the **Start Date and Time**.

When the units is **Minutes** or **Hours**, the job is scheduled to run at the specified time interval. The changing of the date is not significant. If the 24 hours in a day is not evenly divisible by the interval, the job will be submitted at different times on each day.

When the units is **Days** or **Weeks**, the job is scheduled to run at the specified days interval. One Week is the same as 7 days. On each day, the job will be scheduled at the time specified by the **Start Time**.

When the units is **Months**, the job is scheduled to run on the same day of each month. The day of the month is the day specified in the **Start Date**, and the time is the **Start Time**. If the new month does not have the day specified in the **Start Date**, for example, if the start date is 1/31/2005, and the new month is February (there is no February 31), the job will be scheduled in the following month as though the new month had the necessary number of days (in this case, on March 3).

When the units is **Years**, the job is scheduled to run on the same day of each year. The month and day of the year is the month and day specified in the **Start Date**, and the time is the **Start Time**. If the **Start Date** is February 29 of a leap year, and the **Quantity** is not a multiple of 4, then in each non-leap year, the job will be submitted on March 1.

For **Minutes**, **Hours**, and **Days**, the **Quantity** can be a number between 1 and 999. For **Weeks**, **Months** and **Years**, the **Quantity** can be a number between 1 and 99.

Day-of-Week Frequency

A **Day-of-Week** frequencyis used to schedule a job to run on the same day (or days) of each week. More than one day of the week can be selected, with the job being submitted on each of the specified days. On each day, the job will be scheduled at the time specified by the **Start Time**.

If the **Start Date** does not fall on one of the selected days of the week, the job will first be submitted on the next day of the week following the **Start Date** that is selected.

End-of-Month Frequency

An **End-of-Month** frequency is used to schedule a job at the end of each month. The job can be scheduled on the last day of each month, or some number of days before the last day. On each day, the job will be scheduled at the time specified by the **Start Time**.

The number of days before the last day is called the **End-of-Month Days**. The **End-of-Month Days** can be a number between 0 and 27.

Calendar Frequency

A **Calendar** frequency is used to schedule a job to run on any arbitrary set of days you desire. A calendar can specify dates several years in the future. On each day selected by the calendar, the job will be scheduled at the time specified by the **Start Time**.

To use a calendar frequency, you must first define the Calendar to PJS. The Calendar is not a part of the job request, and a single calendar can be specified by many job requests. When you create the calendar, select the dates on which you want the job to run.

Up to three calendars can be specified for job request. If more than one calendar is specified, the job will be submitted only on those days that are specified by all of the calendars.

A Calendar is created with either the **PJCALADD** TSO command, or with the **Add Calendar** ISPF Panel. After a Calendar is created, it can be modified with the **PJCALMOD** TSO command or with the **Modify Calendar** ISPF Panel. When it is no longer needed, it can be deleted with the **PJCALDEL** TSO command or the **Delete Calendar** ISPF panel.

A Calendar is identified by its **Calendar-ID**. The Calendar-ID consists of an **Owner-ID** and a **Calendar Name**. The Calendar Name may be a name chosen by the user.

The Calendar-ID is usually formatted as *owner-ID***.***calendar-name*. A Calendar Name may be any name, 1 to 8 characters long. The first character must be alphabetic or a national character ('$', '#', or '@'). The remaining characters must be alphabetic, numeric, or a national character.

Specifying a Submit Window

Job requests cannot always be processed exactly at the specified run time. A job can be submitted late for several reasons:

- A job can be delayed until all of its job request events have been posted.
- The JCL data set may be in use by another job in the system.
- Your system may be shut down for maintenance.
- Your site may call a halt to job submission for a variety of reasons.
- Normal system processing loads can cause a delay.
- The act of job submission can take time.

Although the last two factors are common, they usually delay PJS job submission by no more than a few minutes. In most cases, a short delay in job submission is not a problem. In some cases, however, submitting a job at the wrong time can cause serious problems.

For example, suppose you want PJS to submit a long-running batch update that enqueues a data set, and suppose that the data set must be available during the day. PJS can schedule the job to be submitted after normal working hours. If, however, system problems cause PJS to be down for an extended period of time, and if the job is submitted late, the job may not finish execution before the data set is needed. One way to remedy this problem is to specify a Submit Window.

A **Submit Window** sets a deadline for job submission. If your job is not submitted before the **Submit Window** expires, you can tell PJS to take one of the following actions:

DISABLE	places the job request in DISABLED status.
ERROR	places the job request in ERROR status.
SKIP	skips this job submission. PJS resets all job request events and recalculates the **Next Run date and Time**.

Using Events to Control Job Submission

In addition to the date and time options mentioned above, you can control job submission with **events**. An event can be used to delay job submission until some outside condition is satisfied. Events are often used to control the submission of dependent jobs.

When the conditions for an event are satisfied, PJS is notified of this fact by either executing the PJSPOST program in a batch job, or manually, using a PJS TSO command or the PJS ISPF Interface. The PJSPOST program can also be executed by an on-line subsystem, or by a system exit. Your PJS System Administrator can tell you if this is done at your installation. When PJS is notified that an event is satisfied, the event is said to be **posted**.

When using events to control job submission, the job request specifies the event (or events) on which it is dependent. The job request will then wait until all the events it specifies have been posted. After all the events are posted, the job can be submitted (if its Next Run Date and Time is past). If one or more events remain unposted, the job will not be submitted.

Creating an Event

Each event defined to PJS has a unique **Event-ID**. The Event-ID consists of an **Owner-ID** and an **Event Name**. The Event Name may be a name chosen by the user.

The Event-ID is usually formatted as *owner-ID***.***event-name*. An Event Name may be any name, 1 to 8 characters long. The first character must be alphabetic or a national character ('$', '#', or '@'). The remaining characters must be alphabetic, numeric, or a national character.

A PJS **Event** is created automatically when a new, unique Event-ID is specified in a job request. If the same Event-ID is specified in another job request, the same event is used. A new event is not created. An event can be specified in any number of job requests.

When an event is not referenced by any job requests it is no longer needed. Unreferenced events are deleted by the PJS Queue Maintenance Utility, which is run periodically by the PJS System Administrator. There is no other way to delete an event.

Job Request Events

A **Job Request Event** is the specification in a job request record that a job request is dependent on an event. In addition to specifying the Event-ID, the Job Request Event also specifies whether the event is posted for this job request. Since each job request that specifies an event keeps its own posting flag, an event may be posted for one job request, but not for another. A job request can specify up to 50 events on which it is dependent.

When an event is posted, PJS will normally post the corresponding Job Request Events in each of the job requests that are dependent on the event. It is also possible to post or reset an individual Job Request Event. This may be done to 'clean up' the job request events after a problem has occurred.

When all of the Job Request Events in a job request indicate that the event has been posted, the job can be submitted. After the job is submitted, all of the Job Request Events in the job request are reset. Only the job request events for the submitted job request are reset. Other job requests that specify the same event are not affected.

Preposted and Non-Preposted Events

By default, a job request event is posted any time the event record is posted, without regard to the scheduled run time of the job request. Job request events can be posted far in advance of the **Next Run Date and Time**. This is called **preposting**.

In some cases, you may not want to post a job request event until after the **Next Run Date and Time** is reached. If the PJS System Task tries to post the job request event before the scheduled run time, the job request event is not posted. The job will be run some time after its scheduled run time, whenever all of its job request events are posted.

Posting an Event

When you tell PJS that the outside condition associated with the event is satisfied, you **post** the event. A PJS Event is usually posted by inserting a step at the appropriate point into the batch job that satisfies the condition.

To post a PJS event from a batch job, you can add the following step to the job:

```
//stepname EXEC PGM=PJSPOST,PARM='owner-id.event-name'
```

Figure 1: PJSPOST JCL

where:

stepname	is any valid stepname.
owner-ID	is the owner-ID of the event to be posted. You must specify this value.
event-name	is the event name of the event to be posted.

For compatibility with release 2.0, the program name **PJSEVENT** can be used as an alias of **PJSPOST**. **PJSPOST** is the recommended program name.

An event can also be posted manually using the **PJEVPOST** TSO command or the **Post Event** ISPF Panel.

If you want to post job request events in a single job request, you can modify the job request using the PJS TSO commands or PJS ISPF panels.

When an event record is posted, PJS puts the event in **POST PENDING** status. The PJS System Task then posts all of the corresponding job request events. After the job request events are posted, the event record returns to its normal state.

Resetting an Event

When you remove the posting from an event, you **reset** the event. It is not normally necessary to reset events explicitly. Job request events are reset automatically when the job is submitted. However, it may be necessary to explicitly reset an event that should not have been posted. To reset an event, you can use one of the following methods:

To reset a PJS event from a batch job, you can add the following step to the job:

```
//stepname EXEC PGM=PJSRESET,PARM='owner-id.event-name'
```

Figure 2: PJSRESET JCL

where:

stepname	is any valid stepname.
owner-ID	is the owner-ID of the event to be posted. You must specify this value.
event-name	is the event name of the event to be posted.

An event can also be reset manually using the **PJEVRSET** TSO command or the **Reset Event** ISPF Panel.

If you want to reset job request events in a single job request, you can modify the job request using the PJS TSO commands or PJS ISPF panels.

When an event record is reset, PJS puts the event in **RESET PENDING** status. The PJS System Task then resets all of the corresponding job request events. After the job request events are reset, the event record returns to its normal state.

Event Status

A **Job Request Event** is either **Posted** or **Not Posted**. A Job Request Event that is posted has a **Date/Time Posted** associated with it. If the Date/Time Posted for a Jopb Request Event has a value, then it is **Posted**. If this date is blank then the Job Request Event is **Not Posted**.

An Event Record is not itself Posted or Not Posted, but can have a Post Pending or Reset Pending status. A status of **POST PENDING** means that the event has been posted, either by the PJSPOST program in a job, or manually by a PJS user, but that fact has not yet been propogated to each Job Request Event by the PJS System Task. Likewise, a status of **RESET PENDING** means that the event has been reset, either by the PJSRESET program in a job, or manually by a PJS user, but that fact has not yet been propogated to each Job Request Event by the PJS System Task. The 'pending' status is short-term, usually no more than a minute.

Modifying the JCL with Variable Substitution

Variable substitution allows you to modify the jobs JCL as it is being submitted. This can be useful for adding parameters to jobs. PJS Global Variables can be used to make it easy to add the same value to several jobs. PJS Dynamic Values can be used to insert PJS or installation defined values that are determined at job submit time.

To use variable substitution, you define to PJS a **Target String** that can be found in your JCL. You then define a **Replacement String** that PJS is to substitute for the Target String in your JCL. Additional information about how to perform the variable substitution can also be specified. A job request can specify up to 20 different variable substitutions. Each substitution will be performed as often as the target string is found in the JCL being submitted.

When variable substitution is performed, each JCL image to be submitted is searched for Target Strings. The input JCL image is searched from the first column to the last. As each column is checked, each target string specified is checked for a match. The target strings are checked in the order specified in the job request. If more than one target string can match at a given column, the first one specified will be used. When a match is found, the target string is replaced by the replacement value specified for that target string. The checking then continues with the column following the replacement value. The replacement value will not be rechecked for other target strings.

The variable substitution takes place only for the JCL submitted to the system. The JCL in the JCL data set or in the PJS JCL Spool is not changed in any way.

Specifying the Target String in the Source JCL

The **Target String** is the character string that PJS is to locate and replace in your JCL. The Target String is a character string up to 8 characters long, and may include special characters and embedded spaces.

Specifying Search Columns

By default, PJS will search an entire JCL image for the target string. If this is not desirable, the search can be limited to specific columns. The **Search Start Column** is the first column that will be searched for the target string. The **Search End Column** is the last column that will be searched.

Specifying the Replacement Value

The **Replacement Value** is the character string that will replace the target string in the JCL. The Replacement Value must resolve to a character string up to 80 characters long, and may include special characters and spaces.

The Replacement Value can be one of three **Replacement Types**:

Literal Value	the replacement value is explicitly specified as a character string.
Global Variable	the replacement value is obtained from a Global Variable.
Dynamic Value	the replacement value is determined when the job is submitted.

Using Literal Values

A **Literal Value** is a character string that is directly inserted into the submitted JCL. The same value will always be inserted.

A Literal Value may be 0 to 80 characters in length, and may specify any EBCDIC character.

Using Global Variables

A **Global Variable** is a replacement value that is specified outside of the job request. Global Variables offer several advantages.

Any number of job requests can use the same global variable. Thus you can specify a parameter value in one place and have it inserted in several job requests.

A Global Variable is created with either the **PJVARADD** TSO command, or with the **Add Global Variable** ISPF Panel. After a Global Variable is created, it can be modified with the **PJVARMOD** TSO command or with the **Modify Global Variable** ISPF Panel. When it is no longer needed, it can be deleted with the **PJVARDEL** TSO command or the **Delete Global Variable** ISPF panel.

Since a Global Variable is a separate entity from the Job Request, if your security system supports it, you can allow another user to update the Global Variable, but not the Job Request, allowing the user to specify a parameter value for a job over which he/she otherwise has no control.

A Global Variable is identified by its **Global Variable-ID** (or simply **Variable-ID** for short). The Variable-ID consists of an **Owner-ID** and an **Variable Name**. The Variable Name may be a name chosen by the user.

The Variable-ID is usually formatted as *owner-ID***.***variable-name*. A Variable Name may be any name, 1 to 8 characters long. The first character must be alphabetic or a national character ('$', '#', or '@'). The remaining characters must be alphabetic, numeric, or a national character.

A Global Variable specifies the replacement value to be used for the variable substitution. This replacement value may be:

- A Literal Value
- A Dynamic Value
- Another Global Variable

A Global Variable may refer to another global variable, which may in turn refer to another global variable. This may continue as often as desired, however, care should be taken that no global variable refers to another global variable that directly or indirectly refers back to itself. If this happens, an error will occur when the PJS attempts to submit the job. When this error occurs, the job submission will be purged, and the Job Request Status will be set to **ERROR**.

Using Dynamic Values

A **Dynamic Value** is a replacement value that is determined by either PJS or an installation provided routine when the job is submitted.

When you use a dynamic value, you specify the Dynamic Value Name. The Dynamic Value Name identifies what type of value is to be used as the replacement value. The value may be defined by PJS, or by an installation provided routine.

PJS currently provides the following Dynamic Values:

SYSDATE is the current system date in the format *mm*/*dd*/*yyyy*.

SYSTIME is the current system time in the format *hh*:*mm*:*ss*.

For a list of any installation provided values, see your **PJS System Administrator**.

Specifying the Data Shift Options

When the length of the replacement string is not the same as the length of the target string, the text following the target string may be adjusted to make the replacement string fit. This can be done in several different ways. The type of shifting desired is called the **Shift Type**. The following shift types are supported:

NONE Characters to the right of the target string are not shifted. If the replacement string is shorter than the target string, the replacement string is padded with spaces. If the replacement string is longer than the target string, it is truncated on the right.

ALL All characters to the right of the target string are shifted. If the replacement string is shorter than the target string, the characters are shifted left. Spaces are inserted on the right of the shifted characters. If the replacement string is longer than the target string, the characters are shifted right. Characters on the right may be truncated.

SPACE If the replacement string is shorter than the target string, the characters to the right of the target string up to the next double spaces are shifted left. If the replacement string is longer than the target string, the characters to the right of the target string are shifted left, replacing double spaces with single spaces, until the replacement string is fit. If insufficient extra spaces are found, the characters are truncated on the right

The columns to be considered for shifting always begin immediately to the right of the target string. The leftmost column of the shifted characters may be limited by the **Shift End Column**. Characters to the right of the Shift End Column are never shifted. If the shifted characters are truncated, they will be truncated at the Shift End Column.

The examples in Figure 3 help show how shifting and truncation occurs:

Assume the following:

Target String - TARGET
Search Start Column - 1
Search End Column - 20
Shift End Column - 43

Replacement String - REPL

Original JCL Statement -

```
         1         2         3         4         5         6         7
123456789012345678901234567890123456789012345678901234567890123456789012
ORIGINAL TARGET JCL  EXAMPLE   STATEMENT    WITH EXTRA TEXT
```

Shift Type - NONE

```
         1         2         3         4         5         6         7
123456789012345678901234567890123456789012345678901234567890123456789012
ORIGINAL REPL   JCL  EXAMPLE   STATEMENT    WITH EXTRA TEXT
```

Shift Type - ALL

```
         1         2         3         4         5         6         7
123456789012345678901234567890123456789012345678901234567890123456789012
ORIGINAL REPL JCL  EXAMPLE   STATEMENT      WITH EXTRA TEXT
```

Shift Type - SPACE

```
         1         2         3         4         5         6         7
123456789012345678901234567890123456789012345678901234567890123456789012
ORIGINAL REPL JCL    EXAMPLE   STATEMENT    WITH EXTRA TEXT
```

Now assume the following:

Replacement String - REPLACEMENT

Shift Type - NONE

```
         1         2         3         4         5         6         7
123456789012345678901234567890123456789012345678901234567890123456789012
ORIGINAL REPLAC JCL  EXAMPLE   STATEMENT    WITH EXTRA TEXT
```

Shift Type - ALL

```
         1         2         3         4         5         6         7
123456789012345678901234567890123456789012345678901234567890123456789012
ORIGINAL REPLACEMENT JCL  EXAMPLE   STATEME WITH EXTRA TEXT
```

Shift Type - SPACE

```
         1         2         3         4         5         6         7
123456789012345678901234567890123456789012345678901234567890123456789012
ORIGINAL REPLACEMENT JCL EXAMPLE STATEMENT  WITH EXTRA TEXT
```

Figure 3: Variable Substitution Shifting Examples

Controlling PJS Messages

While processing a job request, the PJS System Task will normally produce one or more messages that indicate what is being done to the job request. Some of these message simply indicate the status of the job request. Others may indicate errors or other problems encountered in processing the job request.

Although these messages are produced outside the TSO user's address space, and may be produced while the user is not logged on to the system, the user will still frequently need to be aware of them.

Sending Message to a TSO User

PJS will normally attempts to send any messages about a job request to the TSO user responsible for the job request. This is done with the TSO SEND facility.

The TSO user-ID the PJS sends its messages to is called the **Notify User-ID**. By default, this is the Owner-ID of the job request. However, if the Owner-ID is not a TSO User-ID, or if some other user needs to be notified of the status of a job request, this can be specified to be any TSO User-ID desired.

Filtering Messages by Severity Level

When PJS submits a job, several messages can be generated, even when no errors occur. If a user has many job requests, this can become a nuisance. Worse, if many jobs are submitted overnight, when the user logs on to TSO, a 'deluge' of normal PJS messages can be received. the user can get so accustomed to seeing these messages that they are routinely deleted without being read. Then, when an error occurs, the important messages are missed.

To control the volume of messages received, the user can specify a **Notify Message Level**. The Notify Message Level defines the minimum message level that is to be sent to the user. Each Notify Message Level includes the level specified and all more severe levels. Less severe messages will not be sent, however, they can still be retrieved with from the PJS Message History Log.

The following message levels may be specified:

INFO Information messages, plus warnings and errors (all messages)

WARN Warning messages, plus errors

ERROR Error messages only

NONE No messages will be sent

Using the PJS Message History Log

A user will frequently wish to know what happened to a job request. This can happen when a message is suppressed by the Notify Message Level, or if a messages was sent but simply missed or forgotten. The PJS System Task stores most job request messages in the **PJS Message History Log**.

Messages stored in the PJS Message History Log can be retrieved with the **PJREQHST** TSO command, or with the **Display Job Request History** ISPF panel.

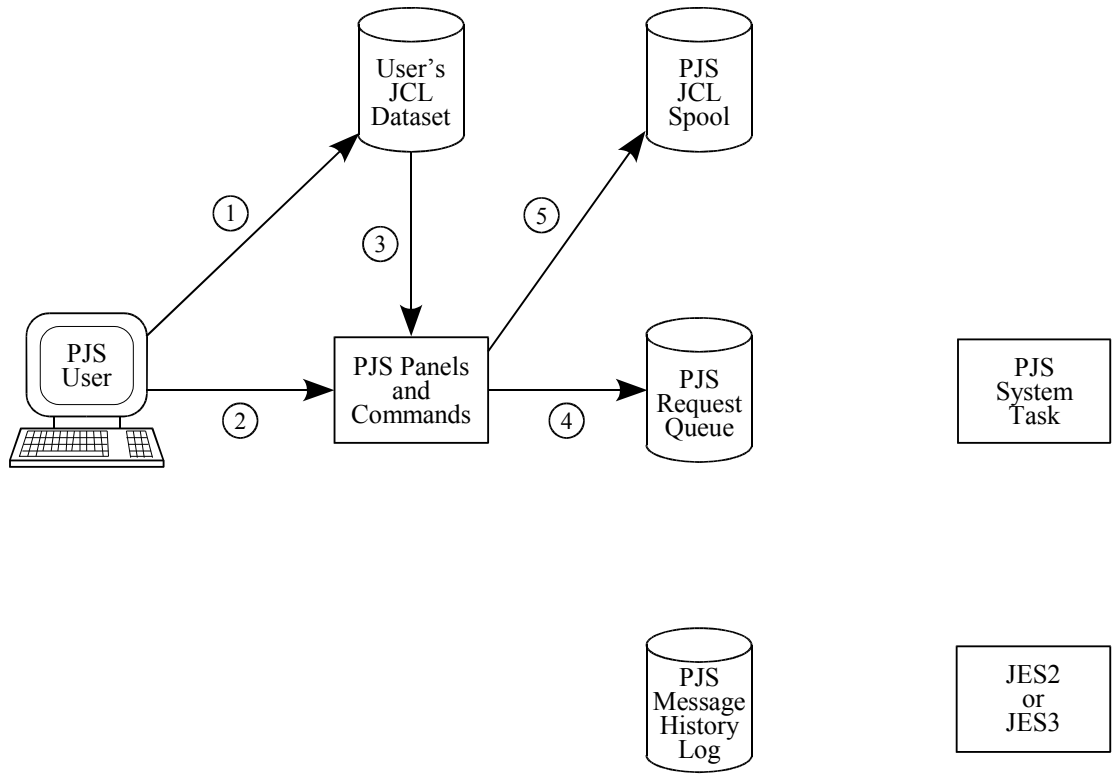
Messages are periodically deleted from the PJS Message History Log based on installation defined criteria.

Chapter 3: PJS Processing Overview

The following sections describe how PJS performs some of its more important functions.

Updating the PJS Database

The following figure illustrates adding a job request to PJS:

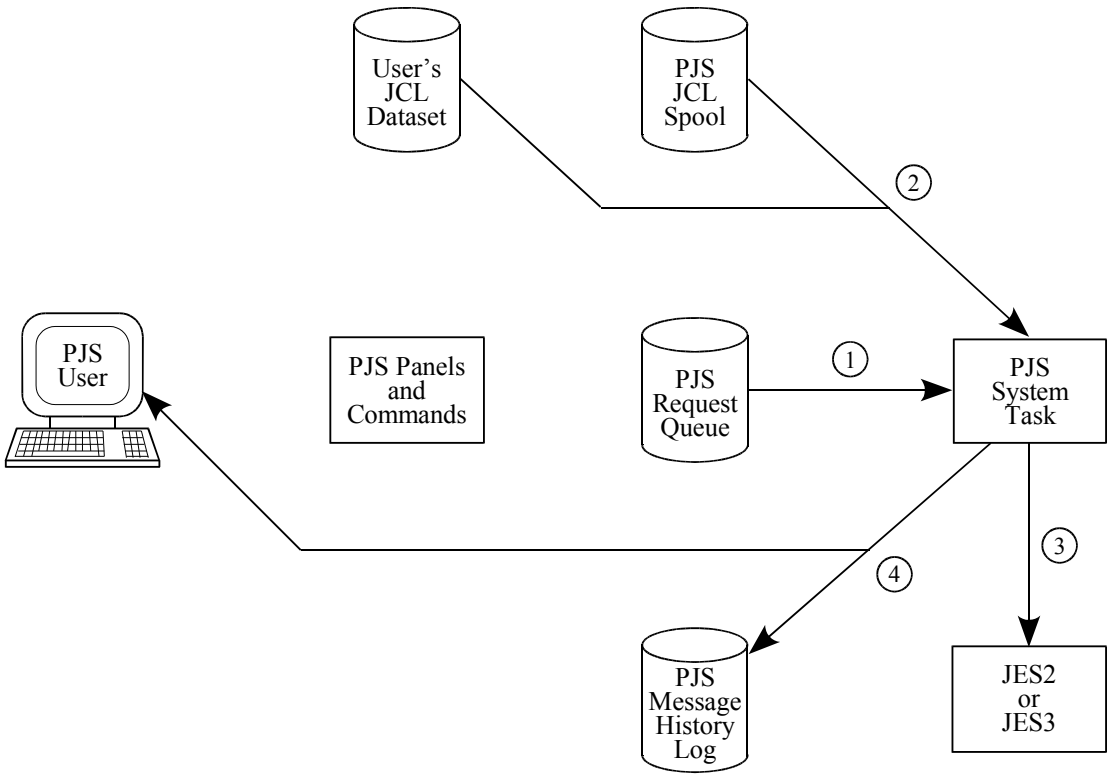


1. The PJS user adds the JCL to be submitted to the User's JCL Data Set using standard TSO commands or ISPF panels.
2. The PJS user adds a Job Request using the PJS TSO commands or ISPF panels.
3. The PJS TSO/ISPF Interface reads the User's JCL Data Set to verify that the JCL exists.
4. The PJS TSO commands or ISPF panels update the PJS Request Queue with the job request.
5. If the job request specifies the JCL Save option, the PJS TSO commands or ISPF panels copy the JCL to the PJS JCL Spool for later submission.

Figure 4: Create Job Request Processing

Submitting a Batch Job

The following figure illustrates how PJS submits a job:

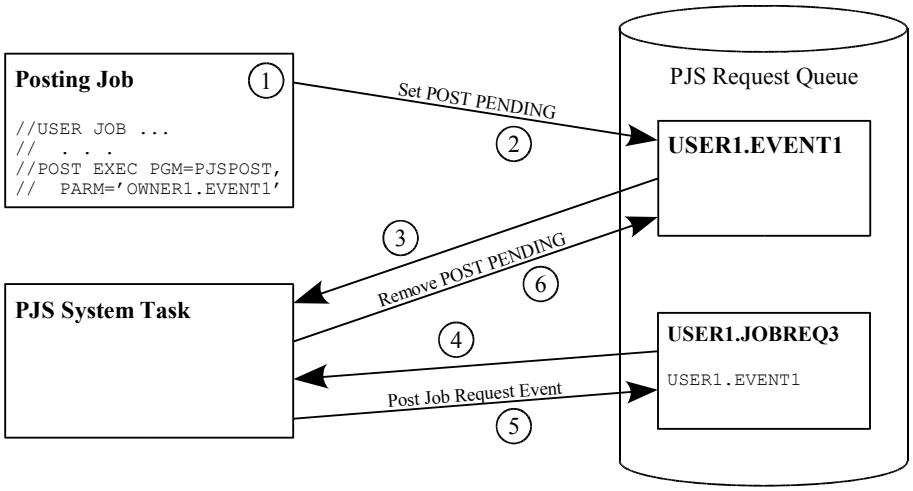


1. The PJS System Task reads the job request record from the PJS Request Queue. It then waits for the Next Run Date/Time. If any job request events are not posted, the PJS System Task continues to wait until all job request events are posted.
2. The JCL is read, either from the User's JCL Data Set, or from the PJS JCL Spool.
3. The JCL written to the JES2 or JES3 internal reader. If variable substitution has been specified, the substitutions are performed as the JCL written.
4. Any messages produced are sent to the TSO user specified by the Notify Userid. Messages are also written to the PJS Message History Log.

Figure 5: Submitting a Batch Job

Posting an Event

The following figure illustrates how PJS posts an event:



1. The job on which other jobs depend includes a step to run the PJSPOST program, usually as the last step in the job. An event can also be posted directly by a TSO/ISPF user.
2. The PJSPOST program updates the event record in the PJS Request Queue, changing its status to "POST PENDING".
3. Within about a minute, the PJS System Task wakes up and sees the event in "POST PENDING" status.
4. The PJS System Task scans the PJS Request Queue, finding off job requests that specify the posted event.
5. The Job Request Event in each of these job requests is posted by recording the current date and time. If the job request event specified "NOPREPOST" it will be posted only if the job request's Next Run Date/Time is past.
6. After posting all job request events, the "POST PENDING" status is removed from the event record.

Figure 6: Posting an Event

Chapter 4: PJS Specification Conventions

Some types of values are used in many PJS commands and parameters. Rather than repeatedly describe the same parameters in detail, this section describes some of these common parameters. These descriptions are then referred to in the detailed descriptions of the TSO commands and ISPF panels where appropriate.

In each of the parameter descriptions, unless otherwise specified, alphabetic refers to the uppercase letters of the alphabet ('A' - 'Z'). Numeric refers to the numbers ('0' - '9'). National refers to the 3 national characters ('$', '#', and '@').

Record-ID Specifications

The Job Request-ID

The **Job Request-ID** (or **Request-ID**) uniquely identifies a job request. A request-ID is specified as:

[*owner-id***.**]*request-name*

where:

owner-id is the owner-ID of the owner of the job request. This may be a name 1 to 8 characters in length. The first character must be an alphabetic or national character. The remaining characters must be alphabetic, numeric, or national characters.

request-name is the name of the job request. This may be a name 1 to 8 characters in length. The first character must be an alphabetic or national character. The remaining characters must be alphabetic, numeric, or national characters. Alternatively, this may also be a 1 to 3 digit number (except 0).

When you input a request-ID it is not necessary to specify the owner-ID. The default is your TSO user-ID. If the default is acceptable you can omit it along with the separating period.

On many ISPF panels the owner-ID and request name are entered in separate fields. If only one field is available, the request-ID is entered just as it is in the TSO commands.

The Calendar-ID

The **Calendar-ID** uniquely identifies a calendar. A calendar-ID is specified as:

owner-id.*calendar-name*

where:

owner-id is the owner-ID of the owner of the calendar. This may be a name 1 to 8 characters in length. The first character must be an alphabetic or national character. The remaining characters must be alphabetic, numeric, or national characters.

calendar-name is the name of the calendar. This may be a name 1 to 8 characters in length. The first character must be an alphabetic or national character. The remaining characters must be alphabetic, numeric, or national characters.

When you input a calendar-ID it is not necessary to specify the owner-ID. The default is your TSO user-ID. If the default is acceptable you can omit it along with the separating period.

On many ISPF panels the owner-ID and calendar name are entered in separate fields. If only one field is available, the calendar-ID is entered just as it is in the TSO commands.

The Event-ID

The **Event-ID** uniquely identifies an event. An event-ID is specified as:

owner-id.*event-name*

where:

owner-id is the owner-ID of the owner of the event. This may be a name 1 to 8 characters in length. The first character must be an alphabetic or national character. The remaining characters must be alphabetic, numeric, or national characters.

event-name is the name of the event. This may be a name 1 to 8 characters in length. The first character must be an alphabetic or national character. The remaining characters must be alphabetic, numeric, or national characters.

When you input a event-ID it is not necessary to specify the owner-ID. The default is your TSO user-ID. If the default is acceptable you can omit it along with the separating period.

On many ISPF panels the owner-ID and event name are entered in separate fields. If only one field is available, the event-ID is entered just as it is in the TSO commands.

The Global Variable-ID

The **Global Variable-ID** (or **Variable-ID**) uniquely identifies a global variable. A variable-ID is specified as:

*owner-id***.***variable-name*

where:

owner-id is the owner-ID of the owner of the global variable. This may be a name 1 to 8 characters in length. The first character must be an alphabetic or national character. The remaining characters must be alphabetic, numeric, or national characters.

variable-name is the name of the global variable. This may be a name 1 to 8 characters in length. The first character must be an alphabetic or national character. The remaining characters must be alphabetic, numeric, or national characters.

When you input a variable-ID it is not necessary to specify the owner-ID. The default is your TSO user-ID. If the default is acceptable you can omit it along with the separating period.

On many ISPF panels the owner-ID and variable name are entered in separate fields. If only one field is available, the variable-ID is entered just as it is in the TSO commands.

Date Specifications

Most dates in PJS may be specified as either an Absolute Date or a Relative Date.

Absolute Dates

An **Absolute Date** is a specific date, for example, 10/22/1992. PJS normally accepts and displays dates in the U.S. format (that is *mm*/*dd*/*yyyy*). Your installation may have changed PJS to use the European date format (that is *dd*/*mm*/*yyyy*). If you are not sure which format your installation uses, check with your PJS Administrator.

If your installation uses the U.S. date format, use the format:

mm/*dd*/*yyyy*

If your installation uses the European date format, user the format:

dd/*mm*/*yyyy*

where:

mm is the 1 or 2 digit month.

dd is the 1 or 2 digit day of the month.

yyyy is the 2 or 4 digit year. If a 2 digit year is entered, the year is assumed to be 20*yy*.

Relative Dates

A **Relative Date** is a date specified as some number of days after the current date. Relative dates can be used to specify a date as "today" or "one week (7 days) from today".

A relative date is specified as:

*+*ddd*

where:

ddd is the 1 to 3 digit number of days to be added to the current date.

Time Specifications

Most times in PJS may be specified as either an Absolute Time or a Relative Time.

Absolute Times

An **Absolute Time** is a specific time, for example, 10:22 AM.

To specify an absolute time, use the format:

hh:*mm* [**AM**|**PM**]

where:

hh is the 1 or 2 digit hour. This must be between 1 and 12 if AM or PM is specified, and between 0 and 23 if neither is specified.

mm is the 1 or 2 digit minutes.

AM indicates that the time is in the morning between midnight and noon.

PM indicated that the time is in the afternoon or evening between noon and midnight.

Relative Times

A **Relative Time** is a time specified as some number of hours and minutes after the current time. Relative times can be used to specify a time as "one hour from now". If a relative time results in a time greater than 11:59:59 PM, the associated date will be incremented to the next day.

A relative time is specified as:

+hh***:***mm*
+hh*
or
+:mm*

where:

hh is the 1 to 2 digit number of hours to be added to the current time.

mm is the 1 to 2 digit number of minutes to be added to the current time.

Chapter 5: The PJS TSO Commands

The PJS TSO commands can be used to create, modify, delete, and list job requests, calendars, events, global variables. PJS TSO commands can be issued from the TSO READY prompt, or anywhere else TSO commands are issues. These commands can be used in TSO CLISTs or REXX EXECs to further automate PJS updates.

PJS Job Request Commands

The PJS Job Request commands enable you to create, modify, delete, and list job requests. The following PJS Job Request commands are available:

PJREQADD is used to create job requests.

PJREQDEL is used to delete job requests.

PJREQHST is used to list messages from the PJS Message History Log for a job request.

PJREQLST is used to list job requests.

PJREQMOD is used to modify job requests.

PJREQADD

The PJREQADD command is used to create a job request.

The PJREQADD command functions as a subcommand processor. When the PJREQADD command is issued, a skeleton job request is created in the TSO user's storage. The user is then prompted to enter subcommands to define the parameters of the job request. As many subcommands as are needed may be entered. The subcommands that may be entered are documented in the section **PJREQADD and PJREQMOD Subcommands** on page 35.

The add is completed when the END subcommand is entered, indicating that all job request parameters have been entered. The PJS Request Queue is not updated in any way until the END subcommand is entered. At any time prior to entering the END subcommand, the user may cancel the add by entering the CANCEL subcommand.

The PJREQADD command has the following format:

```
PJREQADD    [ ID(request-ID) | OWNER(owner-ID) ]
PJRA
```

Figure 7: PJREQADD Command Format

ID(*request-ID***)**
is the request-ID of the job request to be added. The format of the request-ID is described in the section **The Job Request-ID** on page 23. This parameter is mutually exclusive with the **OWNER** parameter.

OWNER(*owner-ID***)**
is the owner-ID of the job request to be added. A numeric request name will be generated. This parameter is mutually exclusive with the **ID** parameter.

If neither the **ID** nor the **OWNER** parameter is entered a request-ID will be generated with the default owner-ID (the TSO user-ID) and a numeric request name.

PJREQDEL

The PJREQDEL command is used to delete a job request.

The PJREQADD command has the following format:

PJREQDEL **PJRD**	*request-ID*

Figure 8: PJREQDEL Command Format

request-ID
is the request-ID of the job request to be deleted. The format of the request-ID is described in the section **The Job Request-ID** on page 23. This parameter is required.

PJREQHST

The PJREQHST command is used to list messages for a job request from the PJS Message History Log.

The PJREQHST command has the following format:

PJREQHST **PJRH**	*request-ID*

Figure 9: PJREQHST Command Format

request-ID
is the request-ID of the job request for which messages are to be listed. The format of the request-ID is described in the section **The Job Request-ID** on page 23. This parameter is required.

PJREQLST

The PJREQLST command is used to display one or more job requests. This command may be used to display a single job request, all job requests for an owner-ID, or all job requests defined to PJS. Only job requests to which the user is authorized will be listed.

The PJREQLST command has the following format:

```
PJREQLST   [ ID(request-ID) | OWNER(owner-ID) | ALL ]
PJRL
           [ SUMMARY | DETAIL ]
```

Figure 10: PJREQLST Command Format

ID(*request-ID*)
is the request-ID of the job request to be displayed. The format of the request-ID is described in the section **The Job Request-ID** on page 23. This parameter is mutually exclusive with the **OWNER** and **ALL** parameters.

OWNER(*owner-ID*)
is the owner-ID for which job requests are to be listed. All job requests for the specified owner-ID to which the user has access will be listed. This parameter is mutually exclusive with the **ID** and **ALL** parameters.

ALL
specifies that all job requests defined to PJS for which the user has access will be listed. This parameter is mutually exclusive with the **ID** and **OWNER** parameters.

If none of the above is specified, the default is to display all job requests for the default owner-ID (the TSO user-ID).

SUMMARY
specifies that only a summary of each job request will be displayed. This is the default when **OWNER** or **ALL** is specified. This parameter is mutually exclusive with the **DETAIL** parameter.

DETAIL
specifies that all information for each job request will be displayed. This is the default when **ID** is specified. This parameter is mutually exclusive with the **SUMMARY** parameter.

PJREQMOD

The PJREQMOD command is used to modify a job request.

The PJREQMOD command functions as a subcommand processor. When the PJREQMOD command is issued, the job request is read from the PJS Request Queue, and updated in the TSO user's storage. The user is then prompted to enter subcommands to change the parameters of the job request. As many subcommands as are needed may be entered The subcommands that may be entered are documented in the section **PJREQADD and PJREQMOD Subcommands** on page 35.

The modify is completed when the END subcommand is entered, indicating that all job request parameters have been entered. The PJS Request Queue is not updated in any way until the END subcommand is entered. At any time prior to entering the END subcommand, the user may cancel the modify by entering the CANCEL subcommand.

The PJREQMOD command has the following format:

```
PJREQMOD    request-ID
PJRM
```

Figure 11: PJREQMOD Command Format

request-ID
is the request-ID of the job request to be modified. The format of the request-ID is described in the section **The Job Request-ID** on page 23. This parameter is required.

PJREQADD and PJREQMOD Subcommands

The following subcommands are available when using the PJREQADD and PJREQMOD commands:

CANCEL	terminates the add or modify without updating the job request.
END	completes the add or modify and updates the job request.
EVENT	specifies parameters for events on which the job request is dependent.
LIST	displays the job request as it has been updated so far.
RESET	clears some basic parameters from the job request.
SET	set some basic parameters in the job request.
VARIABLE	specifies parameters for job request variables.

CANCEL Subcommand

The CANCEL subcommand is used to cancel the add or modify to the job request record. The PJREQADD or PJREQMOD command is terminated without updating PJS.

The CANCEL subcommand has the following format:

```
CANCEL
```

Figure 12: CANCEL Subcommand Format

END Subcommand

The END subcommand is used to complete the add or modify to the job request record. The PJREQADD or PJREQMOD command is terminated after updating PJS.

The END subcommand has the following format:

```
END
```

Figure 13: END Subcommand Format

EVENT Subcommand

The EVENT subcommand is used to specify parameters for the events on which a job request is dependent.

The EVENT subcommand has the following format:

```
EVENT      { CLEAR | ADD | MODIFY | DELETE | POST | RESET }

           event-ID

           [ PREPOST | NOPREPOST ]
```

Figure 14: EVENT Subcommand Format

CLEAR
specifies that all job request events previously specified for the job request are to be deleted. If this parameter is specified, no other parameters may be specified.

ADD
specifies that a job request event is to be added to the job request. This parameter is mutually exclusive with the **CLEAR**, **MODIFY**, **DELETE**, **POST**, and **RESET** parameters.

MODIFY
specifies that a job request event is to be modified. This parameter is mutually exclusive with the **CLEAR**, **ADD**, **DELETE**, **POST**, and **RESET** parameters.

DELETE
specifies that a job request event is to be deleted from the job request. This parameter is mutually exclusive with the **CLEAR**, **ADD**, **MODIFY**, **POST**, and **RESET** parameters.

POST
specifies that a job request event is to be posted. This parameter is mutually exclusive with the **CLEAR**, **ADD**, **MODIFY**, **DELETE**, and **RESET** parameters.

RESET
specifies that a job request event is to be reset. This parameter is mutually exclusive with the **CLEAR**, **ADD**, **MODIFY**, **DELETE**, and **POST** parameters.

One of the above function parameters must be specified immediately following the **EVENT** subcommand name.

event-ID

is the event-ID of the job request event to be added, modified, deleted, posted, or reset in job request. The format of the event-ID is described in the section **The Event-ID** on page 24. This parameter is required except when **CLEAR** is specified.

PREPOST

specifies that the job request event may be posted before the **Next Run Date and Time** for the job request. This is the default when **ADD** is specified. This parameter is valid only when **ADD** or **MODIFY** is specified. This parameter is mutually exclusive with the **NOPREPOST** parameter.

NOPREPOST

specifies that the job request event may not be posted before the **Next Run Date and Time** for the job request. This parameter is valid only when **ADD** or **MODIFY** is specified. This parameter is mutually exclusive with the **NOPREPOST** parameter.

LIST Subcommand

The LIST subcommand is used to display the job request as it has been updated so far.

The LIST subcommand has the following format:

```
LIST
```

Figure 15: LIST Subcommand Format

RESET Subcommand

The RESET subcommand is used to clear some of the basic parameters from the job request.

The RESET subcommand has the following format:

```
RESET      [ENDTIME]

           [RUNTIME]

           [WINDOW]
```

Figure 16: RESET Subcommand Format

ENDTIME
specifies that the **End Date and Time** for the job request is to be cleared. If this parameter is specified, the **Next Run Date and Time** will be recalculated.

RUNTIME
specifies that the **Next Run Date and Time** for the job request is to be cleared and recalculated. The Next Run Date and Time is automatically recalculated only when the **Start Date and Time**, **End Date and Time**, or **Frequency** is changed.

WINDOW
specifies that the Submit Window for the job request is to be cleared.

SET Subcommand

The SET subcommand is used to set some of the basic parameters from the job request.

The SET subcommand has the following format:

```
SET     DESC('description')

       [ENABLED | DISABLED]

        JCLDSN(dsname [ (member) ] )

       [SAVE | NOSAVE | RESAVE]

       [NTFYUSER(user-id) ]

       [NTFYLVL( { INFO | WARN | ERROR | NONE } ) ]

       [STRTDATE(date) ]  STRTTIME(time)

       [ENDDATE(date)  ENDTIME(time) ]

       [WINDOWTIME(time)  [WINDOWOPT( { ERROR | DISABLE | SKIP } ) ] ]

       [  ONCE  |
          YEARS(nn) | MONTHS(nn) | WEEKS(nnn) | DAYS(nnn)  |
          HOURS(nnn) | MINUTES(nnn)  |
          WEEKDAYS( [SUNDAY]  [MONDAY]  [TUESDAY]  [WEDNESDAY]
                    [THURSDAY]  [FRIDAY]  [SATURDAY] )  |
          EOM [ (days) ]  |
          CALENDARS(calendar-ID [ . . . ] )  ]
```

Figure 17: SET Subcommand Format

DESC('*description***')**
specifies the 50-character description of the job request. This is for documentation purposes only and does not affect PJS processing in any way.

ENABLED

specifies that the job request may be processed by PJS. The job request status will be changed to WAIT. This keyword must be used to enable a job request that was placed in the ERROR or HOLD status by the PJS System Task. This parameter is mutually exclusive with the **DISABLED** parameter.

DISABLED

specifies that the job request may not be processed by PJS. The job request status will be changed to DISABLED. PJS will not submit the job for this job request until the job request is reenabled with the ENABLED keyword. This parameter is mutually exclusive with the **ENABLED** parameter.

JCLDSN(*dsname*[**(***member***)**]**)**

specifies the data set name (and member name, if it is a PDS) of the **User's JCL Data Set**. This contains the JCL to be submitted. the data set name and member name should be specified as in most other TSO commands. If the data set name is enclosed in quotes, it is a fully qualified data set name. If it is not enclosed in quotes, the TSO user's data set prefix is included. For an add, this parameter must be specified before the **END** command is entered. If it is not, an error message will be issued and the add will not be completed.

SAVE

specifies that the JCL from the User's JCL Data Set is to be saved in the PJS JCL Spool. When the job is submitted, the saved JCL will be used. Use of this parameter may be restricted by your installation. This parameter is mutually exclusive with the **NOSAVE** and **RESAVE** parameters.

NOSAVE

specifies that the JCL is not to be saved in the PJS JCL Spool. When the job is submitted, the JCL will be obtained directly from the User's JCL Data Set. Use of this parameter may be restricted by your installation. This parameter is mutually exclusive with the **SAVE** and **RESAVE** parameters. This is normally the default, however, the default may be changed by your installation.

RESAVE

specifies that the JCL previously saved in the PJS JCL Spool is to be replaced with a new copy from the User's JCL Data Set. This parameter is mutually exclusive with the **SAVE** and **NOSAVE** parameters.

NTFYUSER(*userid***)**

specifies the TSO user-ID to which the PJS System Task will send messages produced when processing this job request. The default is the owner-ID of the job request.

NTFYLVL(*msglevel*)
specifies the minimum message level to be sent to the TSO user by the PJS System Task. Messages not sent can be listed from the PJS Message History Log, using the **PJREQHST** TSO command, or the **List Job Request History** ISPF panel. Each message level includes all lower levels. The *msglevel* may be one of the following values:

INFO Information messages, plus warnings and errors (all messages). This is the default.

WARN Warning messages, plus errors.

ERROR Error messages only.

NONE No messages will be sent.

STRTDATE(*date*)
specifies the **Start Date** for the job request. The format of the date is described in the section **Date Specifications** on page 25. The default is todays date. If this parameter is specified, the **Next Run Date and Time** will be recalculated.

STRTTIME(*time*)
specifies the **Start Time** for the job request. The format of the time is described in the section **Time Specifications** on page 26. For an add, this parameter must be specified before the **END** command is entered. If it is not, an error message will be issued and the add will not be completed. If this parameter is specified, the **Next Run Date and Time** will be recalculated.

ENDDATE(*date*)
specifies the **End Date** for the job request. The format of the date is described in the section **Date Specifications** on page 25. If this parameter is specified, the **Next Run Date and Time** will be recalculated. To clear the End Date and Time, use the **RESET** subcommand.

ENDTIME(*time*)
specifies the **End Time** for the job request. The format of the time is described in the section **Time Specifications** on page 26. If this parameter is specified, the **Next Run Date and Time** will be recalculated. To clear the End Date and Time, use the **RESET** subcommand.

WINDOWTIME(*time*)
specifies the **Submit Window Time**. If the job cannot be submitted within the *time* specified after the **Next Run Date and Time**, the job wil not be submitted. The action to be taken is specified by the **WINDOWOPT** parameter. The *time* must be in one of the formats:

hh:*mm*
hh
:*mm*

where:

hh is the number of hours after the Next Run date and Time.

mm is the number of minutes after the Next Run Date and Time.

To clear the Submit Window Time and Option, use the **RESET** subcommand.

WINDOWOPT(*option***)**
specifies the Submit Window Option, which is the action to be taken when the Submit Window Time is exceeded. The *option* may be one of the following values:

DISABLE places the job in DISABLED status.

ERROR places the job in ERROR status. This is the default.

SKIP skips this job submission. PJS resets all job request events and recalculates the **Next Run date and Time**.

ONCE
specifies that the job request to be submitted one time only, at the Start Date and Time. This parameter is mutually exclusive with the **YEARS**, **MONTHS**, **WEEKS**, **DAYS**, **HOURS**, **MINUTES**, **WEEKDAYS**, **EOM**, and **CALENDARS** parameters. This is the default.

YEARS(*nn***)**
specifies that the job is to be submitted every *nn* years. *nn* must be a 1 or 2 digit number. For more detailed information on how YEARS scheduling works, see the section **Periodic Frequency** on page 6. This parameter is mutually exclusive with the **ONCE**, **MONTHS**, **WEEKS**, **DAYS**, **HOURS**, **MINUTES**, **WEEKDAYS**, **EOM**, and **CALENDARS** parameters.

MONTHS(*nn***)**
specifies that the job is to be submitted every *nn* months. *nn* must be a 1 or 2 digit number. For more detailed information on how MONTHS scheduling works, see the section **Periodic Frequency** on page 6. This parameter is mutually exclusive with the **ONCE**, **YEARS**, **WEEKS**, **DAYS**, **HOURS**, **MINUTES**, **WEEKDAYS**, **EOM**, and **CALENDARS** parameters.

WEEKS(*nnn***)**
specifies that the job is to be submitted every *nnn* weeks. *nnn* must be a 1 or 2 digit number. For more detailed information on how WEEKS scheduling works, see the section **Periodic Frequency** on page 6. This parameter is mutually exclusive with the **ONCE**, **YEARS**, **MONTHS**, **DAYS**, **HOURS**, **MINUTES**, **WEEKDAYS**, **EOM**, and **CALENDARS** parameters.

DAYS(*nnn*)
specifies that the job is to be submitted every *nnn* days. *nnn* must be a 1 to 3 digit number. For more detailed information on how DAYS scheduling works, see the section **Periodic Frequency** on page 6. This parameter is mutually exclusive with the **ONCE**, **YEARS**, **MONTHS**, **WEEKS**, **HOURS**, **MINUTES**, **WEEKDAYS**, **EOM**, and **CALENDARS** parameters.

HOURS(*nnn*)
specifies that the job is to be submitted every *nnn* hours. *nnn* must be a 1 to 3 digit number. For more detailed information on how HOURS scheduling works, see the section **Periodic Frequency** on page 6. This parameter is mutually exclusive with the **ONCE**, **YEARS**, **MONTHS**, **WEEKS**, **DAYS**, **MINUTES**, **WEEKDAYS**, **EOM**, and **CALENDARS** parameters.

MINUTES(*nnn*)
specifies that the job is to be submitted every *nnn* minutes. *nnn* must be a 1 to 3 digit number. For more detailed information on how MINUTES scheduling works, see the section **Periodic Frequency** on page 6. This parameter is mutually exclusive with the **ONCE**, **YEARS**, **MONTHS**, **WEEKS**, **DAYS**, **HOURS**, **WEEKDAYS**, **EOM**, and **CALENDARS** parameters.

WEEKDAYS(*day* [. . .])
specifies that the job is to be submitted on the specified days of the week. *day* may be a list of one or more of the days of the week (i.e. **SUNDAY**, **MONDAY**, **TUESDAY**, **WEDNESDAY**, **THURSDAY**, **FRIDAY**, and **SATURDAY**). For more detailed information on how WEEKDAYS scheduling works, see the section **Day-of-Week Frequency** on page 7. This parameter is mutually exclusive with the **ONCE**, **YEARS**, **MONTHS**, **WEEKS**, **DAYS**, **HOURS**, **MINUTES**, **EOM**, and **CALENDARS** parameters.

EOM[**(*nn*)**]
specifies that the job is to be submitted at the end of each month. *nn* specifies the number of days before the last day of the month to submit the job. *nn* must be a 1 or 2 digit number between 0 and 27. The default is 0. For more detailed information on how EOM scheduling works, see the section **End-of-Month Frequency** on page 7. This parameter is mutually exclusive with the **ONCE**, **YEARS**, **MONTHS**, **WEEKS**, **DAYS**, **HOURS**, **MINUTES**, **WEEKDAYS**, and **CALENDARS** parameters.

CALENDARS(*cal-ID* [. . .])
specifies that the job is to be submitted on the dates selected by the specified calendars. *cal-ID* is the **Calendar-ID** of the PJS calendar to be used. The format of the calendar-ID is described in the section **The Calendar-ID** on page 24. A list of up to 3 calendars may be specified. For more detailed information on how CALENDARS scheduling works, see the section **Calendar Frequency** on page 7. This parameter is mutually exclusive with the **ONCE**, **YEARS**, **MONTHS**, **WEEKS**, **DAYS**, **HOURS**, **MINUTES**, **WEEKDAYS**, and **EOM**, parameters.

VARIABLE Subcommand

The VARIABLE subcommand is used to specify parameters for the job request variables, which control how the JCL is modified when the job is submitted.

The VARIABLE subcommand has the following format:

```
VARIABLE   { CLEAR | ADD | MODIFY | DELETE }

            TARGET('string'  [seq] )

           [ SEARCH(start   end) ]

           [ SHIFT( { NONE | ALL | SPACE } ) ]

           [ BOUND(end) ]

           { LIT('string') | VAR(var-ID) | DYN(name) }
```

Figure 18: VARIABLE Subcommand Format

CLEAR
specifies that all job request variables previously specified for the job request are to be deleted. If this parameter is specified, no other parameters may be specified.

ADD
specifies that a job request variable is to be added to the job request. This parameter is mutually exclusive with the **CLEAR**, **MODIFY**, and **DELETE** parameters.

MODIFY
specifies that a job request variable is to be modified. This parameter is mutually exclusive with the **CLEAR**, **ADD**, and **DELETE** parameters.

DELETE
specifies that a job request variable is to be deleted from the job request. This parameter is mutually exclusive with the **CLEAR**, **ADD**, and **MODIFY** parameters.

One of the above function parameters must be specified immediately following the **VARIABLE** subcommand name.

TARGET('*string***'** [*seq*]**)**
specifies the target string for the job request variable to be added, modified, or deleted. **'***string***'** is a 1 to 8 character quoted string. *seq* is an optional 1 or 2 digit number specifying which of several job request variables with the same target string is to be modified or deleted. The default is to modify or delete the first job request variable with the specified target string. This parameter is required, except when **CLEAR** is specified.

SEARCH(*start end***)**
specifies the **Search Start and End Columns** for the job request variable. *start* specifies the Search Start Column, and must be a 1 or 2 digit number between 1 and 80. The default is 1. *end* specifies the Search End Column, and must be a 1 or 2 digit number between 1 and 80. The Search End Column must be grerater than or equal to the Search Start Column. The default is 80. This parameter is valid only when **ADD** or **MODIFY** is specified.

SHIFT(*option***)**
specifies the Shift Type for the job request variable. The Shift Type specifies how data to the right of the target string is to be shifted when the length of the replacement string is not the same as the length of target string. For more detailed information on how shifting works, see the section **Specifying the Data Shift Options** on page 14. *option* may be one of the following values:

NONE do not shift.

ALL shift all characters. This is the default.

SPACE shift characters into/from double spaces.

This parameter is valid only when **ADD** or **MODIFY** is specified.

BOUND(*end***)**
specifies the **Shift End Column** for the job request variable. *end* specifies the Shift End Column, and must be a 1 or 2 digit number between 1 and 80. The Shift End Column must be greater than or equal to the Search End Column. The default is 80. This parameter is valid only when **ADD** or **MODIFY** is specified.

LIT('*string***')**
specifies the literal replacement value for the job request variable. **'***string***'** is a 1 to 80 character quoted string. This parameter is valid only when **ADD** or **MODIFY** is specified. This parameter is mutually exclusive with the **VAR** and **DYN** parameters.

VAR(*var-ID***)**
specifies the global variable whose value is to be used as the replacement value for the job request variable. *var-ID* is the **Global Variable-ID** of the PJS global variable to be used. The format of the variable-ID is described in the section **The Global Variable-ID** on page 25. For more detailed

information on how to use global variables, see the section **Using Global Variables** on page 13. This parameter is valid only when **ADD** or **MODIFY** is specified. This parameter is mutually exclusive with the **LIT** and **DYN** parameters.

DYN(*name***)**
specifies the dynamic value name whose value is to be used as the replacement value for the job request variable. *name* specifies the 1 to 16 character name which specifies the dynamic value to be used. Dynamic values may be defined by PJS, or may be provided by an installation provided routine. The dynamic values provided by PJS are described in the section **Using Dynamic Values** on page 14. For a list of any installation provided routines, see your PJS System Administrator. This parameter is valid only when **ADD** or **MODIFY** is specified. This parameter is mutually exclusive with the **LIT** and **VAR** parameters.

When **ADD** is specified, one of the above replacement value parameters (**LIT**, **VAR**, or **DYN**) must be specified.

PJS Calendar Commands

The PJS Calendar commands enable you to create, modify, delete, and list calendars. The following PJS Calendar commands are available:

PJCALADD is used to create calendars.

PJCALDEL is used to delete calendars.

PJCALIST is used to list calendars.

PJCALMOD is used to modify calendars.

PJCALADD

The PJCALADD command is used to create a calendar.

The PJCALADD command has the following format:

```
PJCALADD    calendar-ID
PJCA
            [DESC('description')]

            {DATES(dates [...]) | EXCLUDE(dates [...])}
```

Figure 19: PJCALADD Command Format

calendar-ID
is the Calendar-ID of the calendar to be added. The format of the calendar-ID is described in the section **The Calendar-ID** on page 24. This parameter is required.

DESC('*description***')**
specifies the 50-character description of the calendar. This is for documentation purposes only and does not affect PJS processing in any way.

DATES(*dates* [**...**]**)**
specifies the dates to be selected by the calendar. The dates are entered as a list of single dates and date ranges, separated by spaces or commas. A range of dates is entered as a pair of dates separated by a colon. The format of a date is described in the section **Date Specifications** on page 25. This parameter is mutually exclusive with the **EXCLUDE** parameter.

EXCLUDE(*dates* [**...**]**)**
specifies the dates to be excluded by the calendar. All dates **except** those specified will be selected by the calendar. The dates are entered as a list of single dates and date ranges, separated by spaces or commas. A range of dates is entered as a pair of dates separated by a colon. The format of a date is described in the section **Date Specifications** on page 25. This parameter is mutually exclusive with the **DATES** parameter.

One of the above date parameters (**DATES** or **EXCLUDE**) must be specified.

PJCALDEL

The PJCALDEL command is used to delete a calendar.

The PJCALDEL command has the following format:

PJCALDEL **PJCD**	*calendar-ID*

Figure 20: PJCALDEL Command Format

calendar-ID
is the Calendar-ID of the calendar to be deleted. The format of the calendar-ID is described in the section **The Calendar-ID** on page 24. This parameter is required.

PJCALIST

The PJCALIST command is used to display one or more calendars. This command may be used to display a single calendar, all calendars for an owner-ID, or all calendars defined to PJS. Only calendars to which the user is authorized will be listed.

The PJCALIST command has the following format:

```
PJCALIST   [ID(calendar-ID) | OWNER(owner-ID) | ALL]
PJCL
           [SUMMARY | DETAIL]
```

Figure 21: PJCALIST Command Format

ID(*calendar-ID*)
is the **Calendar-ID** of the calendar to be displayed. The format of the calendar-ID is described in the section **The Calendar-ID** on page 24. This parameter is mutually exclusive with the **OWNER** and **ALL** parameters.

OWNER(*owner-ID*)
is the owner-ID for which calendars are to be listed. All calendars for the specified owner-ID to which the user has access will be listed. This parameter is mutually exclusive with the **ID** and **ALL** parameters.

ALL
specifies that all calendars defined to PJS for which the user has access will be listed. This parameter is mutually exclusive with the **ID** and **OWNER** parameters.

If none of the above is specified, the default is to display all calendars for the default owner-ID (the TSO user-ID).

SUMMARY
specifies that only a summary of each calendar will be displayed. This is the default when **OWNER** or **ALL** is specified. This parameter is mutually exclusive with the **DETAIL** parameter.

DETAIL
specifies that all information for each calendar will be displayed. This is the default when **ID** is specified. This parameter is mutually exclusive with the **SUMMARY** parameter.

PJCALMOD

The PJCALMOD command is used to modify a calendar.

The PJCALMOD command has the following format:

```
PJCALMOD    calendar-ID
PJCM
            [DESC('description')]

            {DATES(dates [...]) | EXCLUDE(dates [...]) |
                        ADDDATES(dates [...]) | DELDATES(dates [...] }
```

Figure 22: PJCALMOD Command Format

calendar-ID
is the Calendar-ID of the calendar to be modified. The format of the calendar-ID is described in the section **The Calendar-ID** on page 24. This parameter is required.

DESC('*description***')**
specifies the 50-character description of the calendar. This is for documentation purposes only and does not affect PJS processing in any way.

DATES(*dates* [**...**]**)**
specifies the dates to be selected by the calendar. The dates are entered as a list of single dates and date ranges, separated by spaces or commas. A range of dates is entered as a pair of dates separated by a colon. The format of a date is described in the section **Date Specifications** on page 25. This parameter is mutually exclusive with the **EXCLUDE**, **ADDDATES**, and **DELDATES** parameters.

EXCLUDE(*dates* [**...**]**)**
specifies the dates to be excluded by the calendar. All dates **except** those specified will be selected by the calendar. The dates are entered as a list of single dates and date ranges, separated by spaces or commas. A range of dates is entered as a pair of dates separated by a colon. The format of a date is described in the section **Date Specifications** on page 25. This parameter is mutually exclusive with the **DATES**, **ADDDATES**, and **DELDATES** parameters.

ADDDATES(*dates* [**. . .**]**)**

specifies the dates to be added to those selected by the calendar. The dates are entered as a list of single dates and date ranges, separated by spaces or commas. A range of dates is entered as a pair of dates separated by a colon. The format of a date is described in the section **Date Specifications** on page 25. This parameter is mutually exclusive with the **DATES**, **EXCLUDE**, and **DELDATES** parameters.

DELDATES(*dates* [**. . .**]**)**

specifies the dates to be deleted from those selected by the calendar. The dates are entered as a list of single dates and date ranges, separated by spaces or commas. A range of dates is entered as a pair of dates separated by a colon. The format of a date is described in the section **Date Specifications** on page 25. This parameter is mutually exclusive with the **DATES**, **EXCLUDE**, and **ADDDATES** parameters.

PJS Event Commands

The PJS Event commands enable you to post, reset, and list events. Events are created and deleted automatically when they are specified in a job request. The following PJS Calendar commands are available:

PJEVLIST is used to list events.

PJEVPOST is used to post events.

PJEVRSET is used to reset events.

PJEVLIST

The PJEVLIST command is used to display one or more events. This command may be used to display a single event, all events for an owner-ID, or all events defined to PJS. Only events to which the user is authorized will be listed.

The PJEVLIST command has the following format:

```
PJEVLIST   [ID(event-ID) | OWNER(owner-ID) | ALL]
PJEL
```

Figure 23: PJEVLIST Command Format

ID(*event-ID***)**
is the **Event-ID** of the event to be displayed. The format of the event-ID is described in the section **The Event-ID** on page 24. This parameter is mutually exclusive with the **OWNER** and **ALL** parameters.

OWNER(*owner-ID***)**
is the owner-ID for which events are to be listed. All events for the specified owner-ID to which the user has access will be listed. This parameter is mutually exclusive with the **ID** and **ALL** parameters.

ALL
specifies that all events defined to PJS for which the user has access will be listed. This parameter is mutually exclusive with the **ID** and **OWNER** parameters.

If none of the above is specified, the default is to display all events for the default owner-ID (the TSO user-ID).

PJEVPOST

The PJEVPOST command is used to post an event. After executing this command the event will be placed in the POST PENDING. The event will remain in this status briefly until the PJS System Task posts all of the corresponding job request events.

The PJEVPOST command has the following format:

```
PJEVPOST    event-ID
PJEP
```

Figure 24: PJEVPOST Command Format

event-ID
is the Event-ID of the event to be posted. The format of the event-ID is described in the section **The Event-ID** on page 24. This parameter is required.

PJEVRSET

The PJEVRSET command is used to reset an event. After executing this command the event will be placed in the RESET PENDING. The event will remain in this status briefly until the PJS System Task resets all of the corresponding job request events.

The PJEVRSET command has the following format:

PJEVRSET **PJER**	*event-ID*

Figure 25: PJEVRSET Command Format

event-ID
is the Event-ID of the event to be reset. The format of the event-ID is described in the section **The Event-ID** on page 24. This parameter is required.

PJS Global Variable Commands

The PJS Global Variable commands enable you to create, modify, delete, and list global variables. The following PJS Global Variable commands are available:

PJVARADD is used to create global variables.

PJVARDEL is used to delete global variables.

PJVARIST is used to list global variables.

PJVARMOD is used to modify global variables.

PJVARADD

The PJVARADD command is used to create a global variable.

The PJVARADD command has the following format:

```
PJVARADD    variable-ID
PJVA
            [DESC('description')]

            {LIT('string') | VAR(var-ID) | DYN(name)}
```

Figure 26: PJVARADD Command Format

variable-ID
is the Variable-ID of the global variable to be added. The format of the variable-ID is described in the section **The Global Variable-ID** on page 25. This parameter is required.

DESC('*description***')**
specifies the 50-character description of the global variable. This is for documentation purposes only and does not affect PJS processing in any way.

LIT('*string***')**
specifies the literal replacement value for the global variable. **'***string***'** is a 1 to 80 character quoted string. This parameter is mutually exclusive with the **VAR** and **DYN** parameters.

VAR(*var-ID***)**
specifies another global variable whose value is to be used as the replacement value for this global variable. *var-ID* is the **Global Variable-ID** of the PJS global variable to be used. This parameter is mutually exclusive with the **LIT** and **DYN** parameters.

DYN(*name***)**
specifies the dynamic value name whose value is to be used as the replacement value for the global variable. *name* specifies the 1 to 16 character name which specifies the dynamic value to be used. Dynamic values may be defined by PJS, or may be provided by an installation provided routine. The dynamic values provided by PJS are described in the section **Using Dynamic Values** on page 14. For a list of any installation provided routines, see your PJS System Administrator. This parameter is mutually exclusive with the **LIT** and **VAR** parameters.

One of the above replacement value parameters (**LIT**, **VAR**, or **DYN**) must be specified.

PJVARDEL

The PJVARDEL command is used to delete a global variable.

The PJVARDEL command has the following format:

PJVARDEL **PJVD**	*variable-ID*

Figure 27: PJVARDEL Command Format

variable-ID
is the Variable-ID of the global variable to be deleted. The format of the variable-ID is described in the section **The Global Variable-ID** on page 25. This parameter is required.

PJVARLST

The PJVARLST command is used to display one or more global variables. This command may be used to display a single global variable, all global variables for an owner-ID, or all global variables defined to PJS. Only global variables to which the user is authorized will be listed.

The PJVARLST command has the following format:

```
PJVARLST   [ID(variable-ID) | OWNER(owner-ID) | ALL]
PJVL
           [SUMMARY | DETAIL]
```

Figure 28: PJVARLST Command Format

ID(*variable-ID*)
is the **Variable-ID** of the global variable to be displayed. The format of the variable-ID is described in the section **The Global Variable-ID** on page 25. This parameter is mutually exclusive with the **OWNER** and **ALL** parameters.

OWNER(*owner-ID*)
is the owner-ID for which global variables are to be listed. All global variables for the specified owner-ID to which the user has access will be listed. This parameter is mutually exclusive with the **ID** and **ALL** parameters.

ALL
specifies that all global variables defined to PJS for which the user has access will be listed. This parameter is mutually exclusive with the **ID** and **OWNER** parameters.

If none of the above is specified, the default is to display all global variables for the default owner-ID (the TSO user-ID).

SUMMARY
specifies that only a summary of each global variable will be displayed. This is the default when **OWNER** or **ALL** is specified. This parameter is mutually exclusive with the **DETAIL** parameter.

DETAIL
specifies that all information for each global variable will be displayed. This is the default when **ID** is specified. This parameter is mutually exclusive with the **SUMMARY** parameter.

PJVARMOD

The PJVARMOD command is used to modify a global variable.

The PJVARMOD command has the following format:

```
PJVARMOD    variable-ID
PJVM
            [DESC('description')]

            [LIT('string') | VAR(var-ID) | DYN(name)]
```

Figure 29: PJVARMOD Command Format

variable-ID
is the Variable-ID of the global variable to be modified. The format of the variable-ID is described in the section **The Global Variable-ID** on page 25. This parameter is required.

DESC('*description***')**
specifies the 50-character description of the global variable. This is for documentation purposes only and does not affect PJS processing in any way.

LIT('*string***')**
specifies the literal replacement value for the global variable. **'***string***'** is a 1 to 80 character quoted string. This parameter is mutually exclusive with the **VAR** and **DYN** parameters.

VAR(*var-ID***)**
specifies another global variable whose value is to be used as the replacement value for this global variable. *var-ID* is the **Global Variable-ID** of the PJS global variable to be used. This parameter is mutually exclusive with the **LIT** and **DYN** parameters.

DYN(*name***)**
specifies the dynamic value name whose value is to be used as the replacement value for the global variable. *name* specifies the 1 to 16 character name which specifies the dynamic value to be used. Dynamic values may be defined by PJS, or may be provided by an installation provided routine. The dynamic values provided by PJS are described in the section **Using Dynamic Values** on page 14. For a list of any installation provided routines, see your PJS System Administrator. This parameter is mutually exclusive with the **LIT** and **VAR** parameters.

Chapter 6: The PJS ISPF Interface

The PJS ISPF Interface consists of menu driven panels that allow the user to quickly and easily update PJS. In addition to the data inquiry and update panels, complete on-line tutorial and help panels are also provided.

General Panels

The PJS Main Menu Panel

The **PJS Main Menu** panel is the first panel displayed when the PJS ISPF interface is entered. The PJS Main Menu panel provides access to the Job Request Menu, the Calendar Menu, the Event Menu, and the Global Variables Menu.

The PJS Main Menu is usually accessed from your ISPF primary option menu. If you are not sure how to access the PJS ISPF interface, see your installations PJS Administrator.

```
yy/mm/dd hh:mm -------  PJS - PERSONAL JOB SCHEDULER  ---------------- 3.1.m
OPTION ===> ____________________________________________________________________

Select one of the following functions:

  J  JOBREQ   - Update Job Requests
  C  CALENDAR - Update Calendars
  E  EVENT    - Update Events
  V  VARIABLE - Update Variables

  T  TUTORIAL - Enter PJS/ISPF Tutorial


Select an option, then press ENTER.


(c) Copyright, Northrop Grumman, 1990, 2005.
For additional copyright information enter ABOUT on the command line.

PJS is distributed under the GNU General Public License.
For license information enter LICENSE on the command line.
```

Figure 30: PJS Main Menu Panel

This panel functions in much the same way as any other ISPF menu panel. Simply enter the letter corresponding to the option desired in the **OPTION** field and press **ENTER**. To return to the previous menu, enter **END** on the option line (or press the **END** PF-key, usually **PF3**).

The following options may be entered on the **OPTION** line:

J selects the **Job Request Menu** panel.

C selects the **Calendar Menu** panel.

E selects the **Event Menu** panel.

V selects the **Global Variable Menu** panel.

T selects the **PJS Tutorial**.

END returns to the calling menu panel (usually the **ISPF Primary Option Menu**).

ABOUT displays detailed copyright information for PJS. This command can be entered from any PJS panel.

LICENSE displays the PJS program license (the GNU General Public License). This command can be entered from any PJS panel.

PJS Tutorial and Help Panels

The **PJS Tutorial and Help** panels provide a complete on-line tutorial and help system. Help panels are provided for each panel in the PJS ISPF interface, and for each updateable field on the data entry panels. Also, a complete Tutorial Index is provided.

The PJS Tutorial is accessed by entering option **T** on the PJS Main Menu panel. The PJS Help panel for a PJS ISPF Interface panel can be accessed by entering the HELP command on the **COMMAND** line (or by pressing the **HELP** PF-key, usually **PF1**).

The PJS Tutorial and Help panels are implemented using the standard ISPF Tutorial and Help system. For detailed information on how to use ISPF Help, see the appropriate ISPF documentation (or enter the **HELP** command again from any of the PJS Tutorial and Help panels.

Job Request Panels

The Job Request panels allow you to add, modify, delete, and display PJS Job Requests.

Job Request Menu Panel

The **Job Request Menu** panel provides access to the job request panels. Inputs are provided for entering the Owner-ID and Request Name of the job request to be updated or displayed.

The **Job Request Menu** panel is accessed from the **PJS Main Menu** panel by selecting option **J**.

```
yy/mm/dd hh:mm -------  PJS - JOB REQUEST MENU  -------------------------------
OPTION ===> ____________________________________________________________________

Select one of the following functions:

  L  LIST     - List all Job Requests for Owner
  A  ADD      - Add a new Job Request
  D  DELETE   - Delete a Job Request
  M  MODIFY   - Modify a Job Request
  S  DISPLAY  - Display a Job Request


Request-ID:
  Owner-ID     ===> ownerid_
  Request Name ===> reqname_   (Required for options D, M, and S)


Enter Request-ID (if required) and make a selection, then press ENTER.
```

Figure 31: Job Request Menu Panel

To use this panel, enter the owner-ID and request name of the job request to be updated or displayed. Then, enter the letter corresponding to the option desired in the **OPTION** field and press **ENTER**. To return to the PJS Main Menu panel, enter **END** on the option line (or press the **END** PF-key, usually **PF3**).

The following options may be entered on the **OPTION** line:

A selects the **Add Job Request** panel.

D selects the **Delete Job Request** panel.

L selects the **List Job Requests** for Owner panel.

M selects the **Modify Job Request** panel.

S selects the **Display Job Request** panel.

END returns to the **PJS Main Menu** panel.

The following record key fields may be entered:

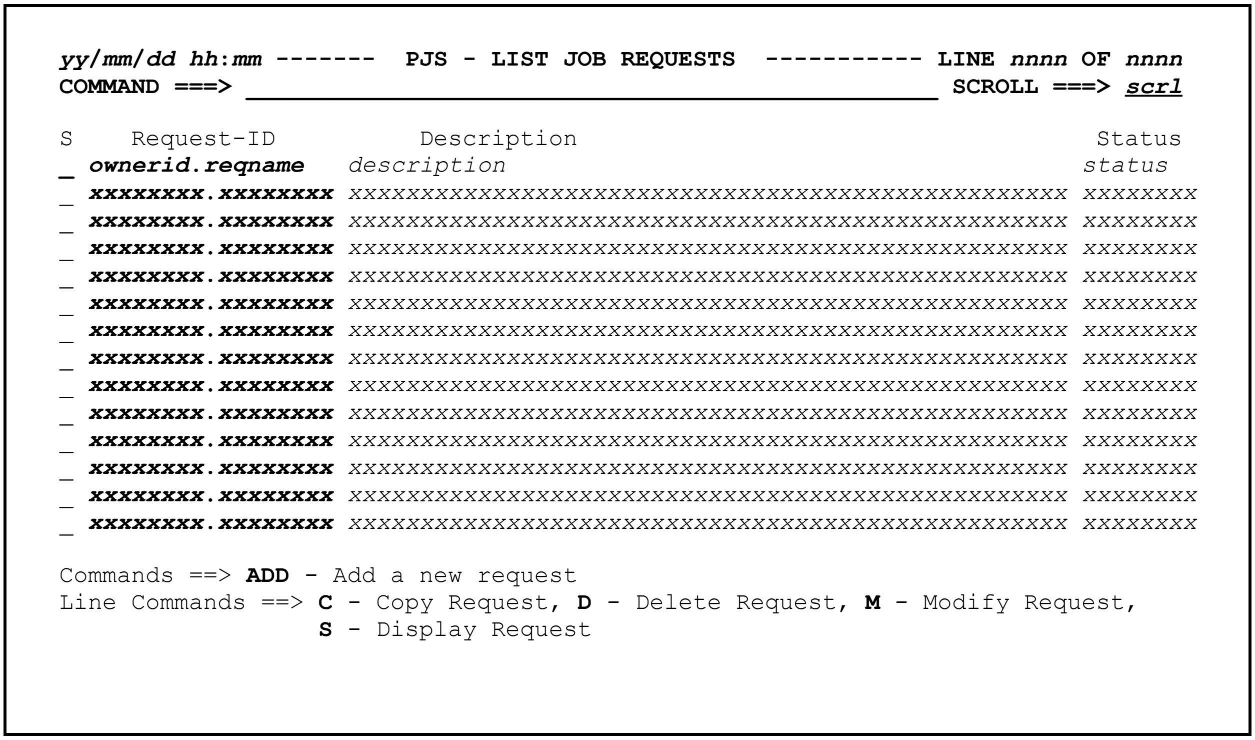
Owner-ID
specifies the **Owner-ID** of the job request(s) to be updated or displayed. This field is required.

Request Name
specifies the **Request Name** of the job request to be updated or displayed. This field is required when option **D** (Delete Job Request), **M** (Modify Job Request), or **S** (Display Job Request) is selected. It is optional when option **A** (Add Job Request) is selected. It is ignored when option **L** (List Job Requests for Owner) is selected.

List Job Requests for Owner Panel

The **List Job Requests for Owner** panel lists the job requests owned by the current Owner-ID. The list can be scrolled up and down to display more job requests, and left and right to display additional information for each job request. Individual job requests may then be selected for update or display using line commands.

The **List Job Requests for Owner** panel is accessed from the **Job Request Menu** panel by selecting option **L**.



```
yy/mm/dd hh:mm -------  PJS - LIST JOB REQUESTS  ----------- LINE nnnn OF nnnn
COMMAND ===> ________________________________________________ SCROLL ===> scrl

S    Request-ID         Description                                      Status
_ ownerid.reqname   description                                        status
_ xxxxxxxx.xxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx
_ xxxxxxxx.xxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx
_ xxxxxxxx.xxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx
_ xxxxxxxx.xxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx
_ xxxxxxxx.xxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx
_ xxxxxxxx.xxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx
_ xxxxxxxx.xxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx
_ xxxxxxxx.xxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx
_ xxxxxxxx.xxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx
_ xxxxxxxx.xxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx
_ xxxxxxxx.xxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx
_ xxxxxxxx.xxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx
_ xxxxxxxx.xxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx

Commands ==> ADD - Add a new request
Line Commands ==> C - Copy Request, D - Delete Request, M - Modify Request,
                  S - Display Request
```

Figure 32: List Job Requests for Owner Panel (Part 1 of 3)

```
yy/mm/dd hh:mm -------  PJS - LIST JOB REQUESTS  ----------- LINE nnnn OF nnnn
COMMAND ===> ________________________________________________ SCROLL ===> scrl

S    Request-ID               JCL Data Set
  ownerid.reqname   data.set.name(member)
_ xxxxxxxx.xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx(xxxxxxxx)
_ xxxxxxxx.xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx(xxxxxxxx)
_ xxxxxxxx.xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx(xxxxxxxx)
_ xxxxxxxx.xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx(xxxxxxxx)
_ xxxxxxxx.xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx(xxxxxxxx)
_ xxxxxxxx.xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx(xxxxxxxx)
_ xxxxxxxx.xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx(xxxxxxxx)
_ xxxxxxxx.xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx(xxxxxxxx)
_ xxxxxxxx.xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx(xxxxxxxx)
_ xxxxxxxx.xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx(xxxxxxxx)
_ xxxxxxxx.xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx(xxxxxxxx)
_ xxxxxxxx.xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx(xxxxxxxx)
_ xxxxxxxx.xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx(xxxxxxxx)

Commands ==> ADD - Add a new request
Line Commands ==> C - Copy Request, D - Delete Request, M - Modify Request,
                  S - Display Request
```

Figure 33: List Job Requests for Owner Panel (Part 2 of 3)

```
yy/mm/dd hh:mm -------  PJS - LIST JOB REQUESTS  ----------- LINE nnnn OF nnnn
COMMAND ===> ________________________________________________ SCROLL ===> scrl

S    Request-ID      Next Date/Time        Frequency        Events   Instdata
_ ownerid.reqname   mm/dd/yyyy hh:mm frequency             events   instdata
_ xxxxxxxx.xxxxxxxx mm/dd/yyyy hh:mm xxxxxxxxxxxxxxxxxxxxx xxxxxxxx xxxxxxxxxxx
_ xxxxxxxx.xxxxxxxx mm/dd/yyyy hh:mm xxxxxxxxxxxxxxxxxxxxx xxxxxxxx xxxxxxxxxxx
_ xxxxxxxx.xxxxxxxx mm/dd/yyyy hh:mm xxxxxxxxxxxxxxxxxxxxx xxxxxxxx xxxxxxxxxxx
_ xxxxxxxx.xxxxxxxx mm/dd/yyyy hh:mm xxxxxxxxxxxxxxxxxxxxx xxxxxxxx xxxxxxxxxxx
_ xxxxxxxx.xxxxxxxx mm/dd/yyyy hh:mm xxxxxxxxxxxxxxxxxxxxx xxxxxxxx xxxxxxxxxxx
_ xxxxxxxx.xxxxxxxx mm/dd/yyyy hh:mm xxxxxxxxxxxxxxxxxxxxx xxxxxxxx xxxxxxxxxxx
_ xxxxxxxx.xxxxxxxx mm/dd/yyyy hh:mm xxxxxxxxxxxxxxxxxxxxx xxxxxxxx xxxxxxxxxxx
_ xxxxxxxx.xxxxxxxx mm/dd/yyyy hh:mm xxxxxxxxxxxxxxxxxxxxx xxxxxxxx xxxxxxxxxxx
_ xxxxxxxx.xxxxxxxx mm/dd/yyyy hh:mm xxxxxxxxxxxxxxxxxxxxx xxxxxxxx xxxxxxxxxxx
_ xxxxxxxx.xxxxxxxx mm/dd/yyyy hh:mm xxxxxxxxxxxxxxxxxxxxx xxxxxxxx xxxxxxxxxxx
_ xxxxxxxx.xxxxxxxx mm/dd/yyyy hh:mm xxxxxxxxxxxxxxxxxxxxx xxxxxxxx xxxxxxxxxxx
_ xxxxxxxx.xxxxxxxx mm/dd/yyyy hh:mm xxxxxxxxxxxxxxxxxxxxx xxxxxxxx xxxxxxxxxxx
_ xxxxxxxx.xxxxxxxx mm/dd/yyyy hh:mm xxxxxxxxxxxxxxxxxxxxx xxxxxxxx xxxxxxxxxxx

Commands ==> ADD - Add a new request
Line Commands ==> C - Copy Request, D - Delete Request, M - Modify Request,
                  S - Display Request
```

Figure 34: List Job Requests for Owner Panel (Part 3 of 3)

For each job request, the following fields are displayed:

Request-ID
is the **Request-ID** of the job request.

Description
is the **Description** of the job request.

Status
is the **Job Request Status** of the job request. For a list of the possible values, see the section **The Job Request Status** on page 4.

JCL Data Set
is the **User's JCL Data Set** name (and member name, if a PDS is specified).

Next Run D/T
is the **Next Run Date and Time** of the job request.

Frequency
is the **Frequency** of the job request. For a list of the possible Frequency values, see the section **Specifying the Frequency** on page 6.

Events
is a summary of the **Job Request Events** specified for the job request. The following vallues may appear:

NONE No events are specified.

POSTED All job request events are currently posted.

xx **OF** *yy* *yy* events are specified for the job request. Of these, *xx* are currently posted.

Instdata
is installation defined data. This is usually used for security related information, and may not be present. For detailed information on your installation data, see your PJS System Administrator.

The following line commands may be entered in the selection field (under the **'S'** on the left side of the panel) for a job request:

C selects the **Add Job Request** panel to add a new job request using the selected job request as a model.

D selects the **Delete Job Request** panel for the selected job request.

M selects the **Modify Job Request** panel for the selected job request.

S selects the **Display Job Request** panel for the selected job request.

The following commands may be entered on the **COMMAND** line:

ADD selects the **Add Job Request** panel for a new job request.

END	returns to the **Job Request Menu** panel.
UP *lines*	scrolls the display up by the specified number of lines.
DOWN *lines*	scrolls the display down by the specified number of lines.
LEFT *cols*	scrolls the display left by the specified number of columns.
RIGHT *cols*	scrolls the display right by the specified number of columns.

Add Job Request Panel

The **Add Job Request** panel is used to add a new job request. When this panel is first displayed, a skeleton job request containing only default information is created in the ISPF user's storage. The user then uses the sub-dialog panels to define the parameters of the job request. The Add Job Request panel displays a summary of the most important parameters.

The add is not completed until the **END** command is entered, indicating that all job request parameters have been entered. At any time prior to entering the END command, the user may cancel the add by entering the **CANCEL** command.

Before a job request can be added, minimal information must be entered using the sub-dialog panels. In particular, a **JCL Data Set Name** must be entered on the **Specify Job Request JCL Source** panel, and a **Start Data and Time** must be entered on the **Specify Job Request Frequency** panel. If these values are not entered when the END command is entered, the appropriate sub-dialog panel will be displayed with a message indicating that required information must be entered.

The **Add Job Request** panel is accessed from the **Job Request Menu** panel by selecting option **A**, or from the **List Job Requests for Owner** panel by either entering the **ADD** command on the command line, or by entering a **C** line command on an existing job request.

If the **Add Job Request** dialog is called from the **List Job Requests for Owner** panel, the **Add Job Request-ID** panel will be displayed first, to prompt the user for the Request-ID to be added.

```
yy/mm/dd hh:mm -------  PJS - ADD JOB REQUEST  --------------------------------
COMMAND ===> ____________________________________________________________________

Request-ID   ===> ownerid.reqname
Description  ===> description_____________________________________

Status       ===> status               Enabled      ===> ___  (Yes or No)

JCL Data Set ===> data.set.name(member)

Next Run D/T ===> mm/dd/yyyy hh:mm
Window       ===> hh:mm  option

Start D/T    ===> mm/dd/yyyy hh:mm     End D/T      ===> mm/dd/yyyy hh:mm

Frequency    ===> frequency

Events       ===> events

Commands ==> JCL - JCL Source, FREQ - Frequency, EVENT - Events,
             OPT - Submit Options, VAR - Variables,
             END - Complete Add, CANCEL - Cancel Add
```

Figure 35: Add Job Request Panel

The following fields are displayed, but may not be updated:

Request-ID
is the **Request-ID** of the job request being added. If a request name was not specified, the request name portion of the request-ID will be '000'. The generated request name will be completed when the add is completed.

Status
is the **Job Request Status** of the job request. For a list of the possible values, see the section **The Job Request Status** on page 4. To update the job request status, use the **Enabled** field on this panel.

JCL Data Set
is the **User's JCL Data Set** name (and member name, if a PDS is specified). To update this parameter, use the **Specify Job Request JCL Source** panel.

Next Run D/T
is the **Next Run Date and Time** of the job request. To update this parameter, use the **Specify Job Request Frequency** panel.

Window
is the **Submit Window Time and Option** of the job request. For a list of the possible Submit Window Option values, see the section **Specifying a Submit Window** on page 8. To update this parameter, use the **Specify Job Request Submit Options** panel.

Start D/T
is the **Start Date and Time** of the job request. To update this parameter, use the **Specify Job Request Frequency** panel.

End D/T
is the **End Date and Time** of the job request. To update this parameter, use the **Specify Job Request Frequency** panel.

Frequency
is the **Frequency** of the job request. For a list of the possible Frequency values, see the section **Specifying the Frequency** on page 6. To update this parameter, use the **Specify Job Request Frequency** panel.

Events
is a summary of the **Job Request Events** specified for the job request. The following vallues may appear:

NONE	No events are specified.
POSTED	All job request events are currently posted.
xx **OF** *yy*	*yy* events are specified for the job request. Of these, *xx* are currently posted.

To update this parameter, use the **Specify Job Request Events** panel.

The following data fields may be updated:

Description
specifies the **Description** of the job request. This is for documentation purposes only and does not affect PJS processing in any way.

Enabled
specifies if the job request can be processed by PJS. If **YES** is entered, the job request status will be changed to **WAIT**. If **NO** is entered, the job request status will be changed to **DISABLED**. **YES** must be entered to enable a job request that was placed in the **ERROR** or **HOLD** status by the PJS System Task.

The following commands may be entered on the **COMMAND** line:

JCL	displays the **Specify Job Request JCL Source** panel.
FREQ	displays the **Specify Job Request Frequency** panel.
EVENT	displays the **Specify Job Request Events** panel.
OPT	displays the **Specify Job Request Submit Options** panel.
VAR	displays the **Specify Job Request Variables** panel.
END	completes the add and returns to the **Job Request Menu** panel, or the **List Job Requests for Owner** panel.

CANCEL cancels the add and returns to the **Job Request Menu** panel, or the **List Job Requests for Owner** panel, without updating PJS.

Add Job Request-ID Panel

The **Add Job Request-ID** panel is used to specify the Job Request-ID when the **Add Job Request** dialog is entered from the **List Job Requests for Owner** panel. This panel is displayed before the **Add Job Request** panel. After this panel is completed, the **Add Job Request** panel will be displayed.

```
yy/mm/dd hh:mm -------  PJS - ADD JOB REQUEST ID  ----------------------------
COMMAND ===> ____________________________________________________________________


New Request-ID:
  Owner-ID     ===> ownerid_
  Request Name ===> reqname_


Enter New Request-ID, then press ENTER.









Commands ==> CANCEL - Cancel Add
```

Figure 36: Add Job Request-ID Panel

The following record key fields may be entered:

Owner-ID
specifies the **Owner-ID** of the job request to be added. This field is required.

Request Name
specifies the **Request Name** of the job request to be added. The format of the request name is described in the section **The Job Request-ID** on page 23. If this field is not entered, a numeric request name will be generated.

The following commands may be entered on the **COMMAND** line:

END returns to the **List Job Requests for Owner** panel.

CANCEL returns to the **List Job Requests for Owner** panel.

Modify Job Request Panel

The **Modify Job Request** panel is used to modify a job request. When this panel is first displayed, the job request is read from the PJS Request Queue and saved in the ISPF user's storage. The user then uses the sub-dialog panels to update the parameters of the job request. The Modify Job Request panel displays a summary of the most important parameters.

The modify is not completed until the **END** command is entered, indicating that all job request parameters have been entered. At any time prior to entering the END command, the user may cancel the modify be entering the **CANCEL** command.

The **Modify Job Request** panel is accessed from the **Job Request Menu** panel by selecting option **M**, or from the **List Job Requests for Owner** panel by entering an **M** line command on an existing request.

```
yy/mm/dd hh:mm -------  PJS - MODIFY JOB REQUEST  ----------------------------
COMMAND ===> ____________________________________________________________________

Request-ID   ===> ownerid.reaqname
Description  ===> description_____________________________________

Status       ===> status               Enabled      ===> ___  (Yes or No)

JCL Data Set ===> data.set.name(member)

Next Run D/T ===> mm/dd/yyyy hh:mm     Last Run D/T ===> mm/dd/yyyy hh:mm:ss
Window       ===> hh:mm  option

Start D/T    ===> mm/dd/yyyy hh:mm     End D/T      ===> mm/dd/yyyy hh:mm

Frequency    ===> frequency

Events       ===> events

Commands ==> JCL - JCL Source, FREQ - Frequency, EVENT - Events,
             OPT - Submit Options, VAR - Variables, HIST - Request History,
             END - Complete Modify, CANCEL - Cancel Modify
```

Figure 37: Modify Job Request Panel

The following fields are displayed, but may not be updated:

Request-ID
is the **Request-ID** of the job request being modified. This parameter cannot be changed.

Status

is the **Job Request Status** of the job request. For a list of the possible values, see the section **The Job Request Status** on page 4. To update the job request status, use the **Enabled** field on this panel.

JCL Data Set

is the **User's JCL Data Set** name (and member name, if a PDS is specified). To update this parameter, use the **Specify Job Request JCL Source** panel.

Next Run D/T

is the **Next Run Date and Time** of the job request. To update this parameter, use the **Specify Job Request Frequency** panel.

Last Run D/T

is the **Last Run Date and Time** of the job request. This is for documentation purposes only and does not affect PJS processing in any way.

Window

is the **Submit Window Time and Option** of the job request. For a list of the possible Submit Window Option values, see the section **Specifying a Submit Window** on page 8. To update this parameter, use the **Specify Job Request Submit Options** panel.

Start D/T

is the **Start Date and Time** of the job request. To update this parameter, use the **Specify Job Request Frequency** panel.

End D/T

is the **End Date and Time** of the job request. To update this parameter, use the **Specify Job Request Frequency** panel.

Frequency

is the **Frequency** of the job request. For a list of the possible Frequency values, see the section **Specifying the Frequency** on page 6. To update this parameter, use the **Specify Job Request Frequency** panel.

Events

is a summary of the **Job Request Events** specified for the job request. The following vallues may appear:

NONE	No events are specified.
POSTED	All job request events are currently posted.
xx **OF** *yy*	*yy* events are specified for the job request. Of these, *xx* are currently posted.

To update this parameter, use the **Specify Job Request Events** panel.

The following data fields may be updated:

Description
specifies the **Description** of the job request. This is for documentation purposes only and does not affect PJS processing in any way.

Enabled
specifies if the job request can be processed by PJS. If **YES** is entered, the job request status will be changed to **WAIT**. If **NO** is entered, the job request status will be changed to **DISABLED**. **YES** must be entered to enable a job request that was placed in the **ERROR** or **HOLD** status by the PJS System Task.

The following commands may be entered on the **COMMAND** line:

JCL	displays the **Specify Job Request JCL Source** panel.
FREQ	displays the **Specify Job Request Frequency** panel.
EVENT	displays the **Specify Job Request Events** panel.
OPT	displays the **Specify Job Request Submit Options** panel.
VAR	displays the **Specify Job Request Variables** panel.
HIST	displays the **Display Job Request History** panel.
END	completes the modify and returns to the **Job Request Menu** panel, or the **List Job Requests for Owner** panel.
CANCEL	cancels the modify and returns to the **Job Request Menu** panel, or the **List Job Requests for Owner** panel, without updating PJS.

Delete Job Request Panel

The **Delete Job Request** panel is used to delete a job request. To confirm the delete, the user must enter **YES** in the confirmation field. The delete is then completed by entering the **END** command. The user may cancel the delete by entering the **CANCEL** command.

The **Delete Job Request** panel is accessed from the **Job Request Menu** panel by selecting option **D**, or from the **List Job Requests for Owner** panel by entering a **D** line command on an existing request.

```
yy/mm/dd hh:mm -------  PJS - DELETE JOB REQUEST  ----------------------------
COMMAND ===> ____________________________________________________________________

Request-ID   ===> ownerid.reqname      Status       ===> status
Description  ===> description

JCL Data Set ===> data.set.name(member)

Next Run D/T ===> mm/dd/yyyy hh:mm     Last Run D/T ===> mm/dd/yyyy hh:mm:ss

Start D/T    ===> mm/dd/yyyy hh:mm     End D/T      ===> mm/dd/yyyy hh:mm

Frequency    ===> frequency

Events       ===> events

Enter 'YES' and press enter to confirm delete ===> ___


Commands ==> EVENT - Events, VAR - Variables, HIST - Request History,
             JCL - Browse JCL,
             END - Complete Delete, CANCEL - Cancel Delete
```

Figure 38: Delete Job Request Panel

The following fields are displayed, but may not be updated:

Request-ID
is the **Request-ID** of the job request being deleted.

Status
is the **Job Request Status** of the job request. For a list of the possible values, see the section **The Job Request Status** on page 4.

Description
specifies the **Description** of the job request.

JCL Data Set
is the **User's JCL Data Set** name (and member name, if a PDS is specified).

Next Run D/T
is the **Next Run Date and Time** of the job request.

Last Run D/T
is the **Last Run Date and Time** of the job request.

Start D/T
is the **Start Date and Time** of the job request.

End D/T
is the **End Date and Time** of the job request.

Frequency
is the **Frequency** of the job request. For a list of the possible Frequency values, see the section **Specifying the Frequency** on page 6.

Events
is a summary of the **Events** specified for the job request. The following vallues may appear:

NONE	No events are specified.
POSTED	All job request events are currently posted.
xx **OF** *yy*	*yy* events are specified for the job request. Of these, *xx* are currently posted.

The following commands may be entered on the **COMMAND** line:

EVENT	displays the **Display Job Request Events** panel.
VAR	displays the **Display Job Request Variables** panel.
HIST	displays the **Display Job Request History** panel.
JCL	displays the JCL to be submitted, either from the user's JCL data set, or the PJS JCL Spool, using ISPF Browse.
END	completes the delete and returns to the **Job Request Menu** panel, or the **List Job Requests for Owner** panel.
CANCEL	cancels the delete and returns to the **Job Request Menu** panel, or the **List Job Requests for Owner** panel, without updating PJS.

Display Job Request Panel

The **Display Job Request** panel is used to display a job request.

The **Display Job Request** panel is accessed from the **Job Request Menu** panel by selecting option **S**, or from the **List Job Requests for Owner** panel by entering an **S** line command on an existing request.

```
yy/mm/dd hh:mm -------  PJS - DISPLAY JOB REQUEST  ----------------------------
COMMAND ===> ____________________________________________________________________

Request-ID   ===> ownerid.reqname      Status       ===> status
Description  ===> description

JCL Data Set ===> data.set.name(member)

Next Run D/T ===> mm/dd/yyyy hh:mm     Last Run D/T ===> mm/dd/yyyy hh:mm:ss
Window       ===> hh:mm  option

Start D/T    ===> mm/dd/yyyy hh:mm     End D/T      ===> mm/dd/yyyy hh:mm

Frequency    ===> frequency

Events       ===> events



Commands ==> EVENT - Events, VAR - Variables, HIST - Request History,
             JCL - Browse JCL
```

Figure 39: Display Job Request Panel

The following fields are displayed:

Request-ID
is the **Request-ID** of the job request being modified.

Status
is the **Job Request Status** of the job request. For a list of the possible values, see the section **The Job Request Status** on page 4.

Description
is the **Description** of the job request.

JCL Data Set
is the **User's JCL Data Set** name (and member name, if a PDS is specified).

Next Run D/T
is the **Next Run Date and Time** of the job request.

Last Run D/T
is the **Last Run Date and Time** of the job request.

Window
is the **Submit Window Time and Option** of the job request. For a list of the possible Submit Window Option values, see the section **Specifying a Submit Window** on page 8.

Start D/T
is the **Start Date and Time** of the job request.

End D/T
is the **End Date and Time** of the job request.

Frequency
is the **Frequency** of the job request. For a list of the possible Frequency values, see the section **Specifying the Frequency** on page 6.

Events
is a summary of the **Job Request Events** specified for the job request. The following vallues may appear:

NONE	No events are specified.
POSTED	All job request events are currently posted.
xx **OF** *yy*	*yy* events are specified for the job request. Of these, *xx* are currently posted.

The following commands may be entered on the **COMMAND** line:

EVENT	displays the **Display Job Request Events** panel.
VAR	displays the **Display Job Request Variables** panel.
HIST	displays the **Display Job Request History** panel.
JCL	displays the JCL to be submitted, either from the user's JCL data set, or the PJS JCL Spool, using ISPF Browse.
END	returns to the **Job Request Menu** panel, or the **List Job Requests for Owner** panel.

Specify Job Request JCL Source Panel

The **Specify Job Request JCL Source** panel is used to enter information about the User's JCL Data Set when adding or modifying a job request.

The **Specify Job Request JCL Source** panel is accessed from the **Add Job Request** panel or the **Modify Job Request** panel by entering the **JCL** command on the command line.

```
yy/mm/dd hh:mm -------  PJS - SPECIFY JCL SOURCE  ----------------------------
COMMAND ===> ____________________________________________________________________

Request-ID   ===> ownerid.reqname      Status       ===> status
Description  ===> description


JCL Data Set Name ===> data.set.name_____________________________
    Member Name   ===> member__

JCL Save Option   ===> option_      (Yes, No, or Refresh)





Commands ==> BROWSE - Browse JCL, BRDATA - Browse JCL Data Set,
             FREQ - Frequency, EVENT - Events, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
```

Figure 40: Specify Job Request JCL Source Panel

The following fields are displayed:

Request-ID
is the **Request-ID** of the job request being modified.

Status
is the **Job Request Status** of the job request. For a list of the possible values, see the section **The Job Request Status** on page 4.

Description
is the **Description** of the job request.

The following data fields may be updated:

JCL Data Set Name
specifies the data set name of the **User's JCL Data Set**. Enter a fully qualified data set name without quotes and without a member name. This parameter is required.

Member Name
specifies the member name of the **User's JCL Data Set**. This parameter is required when the User's JCL Data Set is a PDS. Otherwise, it must be omitted.

JCL Save Option
specifies if the JCL from the User's JCL Data Set is to be saved in the PJS JCL Spool. The following values may be specified:

YES	indicates that the JCL will be saved in the PJS JCL Spool. When the job is submitted, the saved JCL will be used.
NO	indicates that the JCL will not be saved in the PJS JCL Spool. When the job is submitted, the JCL will be obtained directly from the user's JCL data set.
REFRESH	indicates that the JCL previously saved in the PJS JCL Spool is to be replaced with a new copy from the user's JCL data set.

Use of this parameter may be restricted by your installation.

The following commands may be entered on the **COMMAND** line:

BROWSE	displays the JCL to be submitted, either from the user's JCL data set, or the PJS JCL Spool, using ISPF Browse.
BRDATA	displays the JCL from the user's JCL data set (even if the JCL has been saved in the PJS JCL Spool), using ISPF Browse.
FREQ	displays the **Specify Job Request Frequency** panel.
EVENT	displays the **Specify Job Request Events** panel.
OPT	displays the **Specify Job Request Submit Options** panel.
VAR	displays the **Specify Job Request Variables** panel.
HIST	displays the **Display Job Request History** panel.
END	returns to the **Add Job Request** panel or the **Modify Job Request** panel.
CANCEL	cancels the add or modify and returns to the **Job Request Menu** panel, or the **List Job Requests for Owner** panel, without updating PJS.

Specify Job Request Frequency Panel

The **Specify Job Request Frequency** panel is used to enter information about when the job is to be submitted.

The **Specify Job Request Frequency** panel is accessed from the **Add Job Request** panel or the **Modify Job Request** panel by entering the **FREQ** command on the command line.

```
yy/mm/dd hh:mm -------  PJS - SPECIFY JOB FREQUENCY  -------------------------
COMMAND ===> ____________________________________________________________________

Request-ID   ===> ownerid.reqname      Status       ===> status
Description  ===> description

Next Run D/T ===> mm/dd/yyyy hh:mm     Recalculate  ===> ___  (Yes or No)

Start D/T    ===> mm/dd/yyyy hh:mm___  End D/T      ===> mm/dd/yyyy hh:mm___

Frequency (choose one of the options below)
   Periodic     ===> nnn (num) units__ (units - Yr, Mo, Wk, Day, Hr, or Min)
      or
   Day of Week  ===> s Sun  s Mon  s Tue  s Wed  s Thu  s Fri  s Sat
      or
   End of Month ===> Last Day - nn   (days before last day of each month)
      or
   Calendar(s)  ===> ownerid.calname__ ownerid.calname__ ownerid.calname__
      or
   Once (if none of the above is specified)

Commands ==> JCL - JCL Source, EVENT - Events, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
```

Figure 41: Specify Job Request Frequency Panel

The following fields are displayed, but may not be updated:

Request-ID
is the **Request-ID** of the job request being modified.

Status
is the **Job Request Status** of the job request. For a list of the possible values, see the section **The Job Request Status** on page 4.

Description
is the **Description** of the job request.

Next Run D/T
is the **Next Run Date and Time** of the job request. To update the Next Run Date and Time, use the **Recalculate** field on this panel.

The following data fields may be updated:

Recalculate
specifies if the **Next Run Date and Time** is to be recalculated. The Next Run date and Time is automatically recalculated when the **Start Date and Time**, **End Date and Time**, or **Frequency** is changed. Enter **YES** to force the recalculation without changing these parameters.

Start D/T
specifies the **Start Date and Time** of the job request. The format of the date is described in the section **Date Specifications** on page 25. The format of the time is described in the section **Time Specifications** on page 26. This parameter is required.

End D/T
specifies the **End Date and Time** of the job request. The format of the date is described in the section **Date Specifications** on page 25. The format of the time is described in the section **Time Specifications** on page 26.

The **Frequency** specifies how often the job is to be submitted. Several types of frequencies are supported by PJS. Only one may be sepecified. Each of the frequency types is described in detail in the section **Specifying the Frequency** on page 6. Only one of the following frequency types may be specified:

Periodic
specifies that the job is to be submitted at a regular interval. The **Periodic Quantity and Units** specify the interval. The Periodic Quantity may be a number between 1 and 999. The **Periodic Units** may be one of the following:

YEARS specifies that the Periodic Quantity represents years. This may also be specified as **YR**. The Periodic Quantity must be between 1 and 99.

MONTHS specifies that the Periodic Quantity represents months. The Periodic Quantity must be between 1 and 99.

WEEKS specifies that the Periodic Quantity represents weeks. This may also be specified as **WK**.

DAYS specifies that the Periodic Quantity represents days.

HOURS specifies that the Periodic Quantity represents hours. This may also be specified as **HR**.

MINUTES specifies that the Periodic Quantity represents minutes.

Day of Week

specifies that the job is to be submitted on the selected days of the week. The days to be selected may be specified by entering any non-blank character in the unprotected field before the day of the week. A day of the week may be omitted by entering a space in the unprotected field before the day of the week.

End of Month

specifies that the job is to be submitted at the end of each month. The End of Month Days must be a number between 0 and 27. The job is submitted the specified number of days before the last day of the month.

Calendar(s)

specifies that the job is to be submitted on the dates selected by the specified calendars. The format of the calendar-ID is described in the section **The Calendar-ID** on page 24. A list of up to 3 Calendar-IDs may be specified.

Once

if no parameters are specified for any of the above frequency types, the default frequency of **Once** will be assigned.

The following commands may be entered on the **COMMAND** line:

JCL	displays the **Specify Job Request JCL Source** panel.
EVENT	displays the **Specify Job Request Events** panel.
OPT	displays the **Specify Job Request Submit Options** panel.
VAR	displays the **Specify Job Request Variables** panel.
HIST	displays the **Display Job Request History** panel. This command is valid only when this panel was called from the **Modify Job Request** panel.
END	returns to the **Add Job Request** panel or the **Modify Job Request** panel.
CANCEL	cancels the add or modify and returns to the **Job Request Menu** panel, or the **List Job Requests for Owner** panel, without updating PJS.

Specify Job Request Submit Options Panel

The **Specify Job Request Submit Options** panel is used to enter miscelaneous submit options, including the Submit Window and the message processing options.

The **Specify Job Request Submit Options** panel is accessed from the **Add Job Request** panel or the **Modify Job Request** panel by entering the **OPT** command on the command line.

```
yy/mm/dd hh:mm -------  PJS - SPECIFY SUBMIT OPTIONS  -------------------------
COMMAND ===> ____________________________________________________________________

Request-ID   ===> ownerid.reqname      Status       ===> status
Description  ===> description


Window Time      ===> hh:mm        (hh:mm)
Window Optn      ===> option__     (Disable, Error, or Skip)

Notify Userid    ===> userid__
Notify Level     ===> msglevel     (Info, Warn, Error, or None)


                  installation data heading
Inst Data    ===> installation data



Commands ==> JCL - JCL Source, FREQ - Frequency, EVENT - Events,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
```

Figure 42: Specify Job Request Submit Options Panel

The following fields are displayed, but may not be updated:

Request-ID
is the **Request-ID** of the job request being modified.

Status
is the **Job Request Status** of the job request. For a list of the possible values, see the section **The Job Request Status** on page 4.

Description
is the **Description** of the job request.

Inst Data

is installation defined data. This is usually used for security related information, and may not be present. For detailed information on your installation data, see your PJS System Administrator.

The following data fields may be updated:

Window Time

specifies the **Submit Window Time**. If the job cannot be submitted within the *time* specified after the **Next Run date and Time**, the job will not be submitted. The action to be taken is specified by the Submit Window Option parameter. The time must be in the format:

hh:*mm*
hh
:*mm*

where:

hh is the number of hours after the Next Run date and Time.

mm is the number of minutes after the Next Run Date and Time.

To clear the Submit Window Time and Option, use the **RESET** subcommand.

Window Option

specifies the **Submit Window Option**, which is the action to be taken when the **Submit Window Time** is exceeded. The **Submit Window Option** may be one of the following values:

DISABLE places the job in DISABLED status.

ERROR places the job in ERROR status.

SKIP skips this job submission. PJS resets all job request events and recalculates the **Next Run date and Time**.

Notify Userid

specifies the TSO User-ID to which the PJS System task will send messages produced when processing the job request.

Notify Level

specifies the minimum message level to be sent to the TSO user by the PJS System Task. Messages not sent can be listed from the PJS Message History Log, using the **PJREQHST** TSO command, or the **List Job Request History** ISPF panel. Each message level includes all lower levels. The **Notify Message Level** may be one of the following values:

INFO Information messages, plus warnings and errors (all messages).

WARN Warning messages, plus errors.

ERROR Error messages only.

NONE No messages will be sent.

The following commands may be entered on the **COMMAND** line:

JCL displays the **Specify Job Request JCL Source** panel.

FREQ displays the **Specify Job Request Frequency** panel.

EVENT displays the **Specify Job Request Events** panel.

VAR displays the **Specify Job Request Variables** panel.

HIST displays the **Display Job Request History** panel. This command is valid only when this panel was called from the **Modify Job Request** panel.

END returns to the **Add Job Request** panel or the **Modify Job Request** panel.

CANCEL cancels the add or modify and returns to the **Job Request Menu** panel, or the **List Job Requests for Owner** panel, without updating PJS.

Specify Job Request Events Panel

The **Specify Job Request Events** panel is used to enter and update job request events. The list of job request events can be scrolled up and down to display more job request events. Existing job request events may then be selected for update using line commands. At the end of the list of existing job request events are blank entries that may be used for adding new job request events.

The **Specify Job Request Events** panel is accessed from the **Add Job Request** panel or the **Modify Job Request** panel by entering the **EVENT** command on the command line.

```
yy/mm/dd hh:mm ------  PJS - SPECIFY JOB REQUEST EVENTS  --- LINE nnnn OF nnnn
COMMAND ===> ____________________________________________________ SCROLL ===> scrl

Request-ID   ===> ownerid.reqname      Status       ===> status
Description  ===> description

S      Event-ID     Prepost  Date/Time Posted
_  ownerid.evntname   xxx    mm/dd/yyyy hh:mm:ss
_  xxxxxxxx.xxxxxxxx  xxx    mm/dd/yyyy hh:mm:ss
_  xxxxxxxx.xxxxxxxx  xxx    mm/dd/yyyy hh:mm:ss
_  xxxxxxxx.xxxxxxxx  xxx    mm/dd/yyyy hh:mm:ss
_  xxxxxxxx.xxxxxxxx  xxx    mm/dd/yyyy hh:mm:ss
_  ________________   ___
_  ________________   ___
_  ________________   ___
_  ________________   ___
_  ________________   ___

Commands ==> JCL - JCL Source, FREQ - Frequency, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
Line Commands ==> D - Delete Event, P - Post Job Event, R - Reset Job Event
```

Figure 43: Specify Job Request Events Panel

The following fields are displayed, but may not be updated:

Request-ID
is the **Request-ID** of the job request being modified.

Status
is the **Job Request Status** of the job request. For a list of the possible values, see the section **The Job Request Status** on page 4.

Description
is the **Description** of the job request.

The following line commands may be entered in the selection field (under the **'S'** on the left side of the panel) for a job request event:

D deletes the job request event.

P posts the job request event.

R resets the job request event.

For each job request event, the following data fields may be updated:

Event-ID
specifies the **Event-ID** of the job request event. This field may be entered only for a new job request event. Once the Event-ID has been entered and accepted, it will be protected, and cannot be changed. To change an Event-ID, the old job requests event musts be deleted and the new event-ID added. The format of the event-ID is described in the section **The Event-ID** on page 24.

Prepost
specifies if the job request event may be posted before the **Next Run Date and Time** for the job request. **YES** means that the job request event can ne posted at any time. **NO** means that the job request evetn can only be posted if the **Next Run Date and Time** is past. The default is **YES**.

For each job request event, the following fields are displayed, but may not be updated:

Date/Time Posted
If the job request event is currently posted, this is the date and time the job request event was posted. If the job request event is not currently posted, this field is blank. To update the Date and Time Posted, use the **P** or **R** line command on this panel.

The following commands may be entered on the **COMMAND** line:

JCL displays the **Specify Job Request JCL Source** panel.

FREQ displays the **Specify Job Request Frequency** panel.

OPT displays the **Specify Job Request Submit Options** panel.

VAR displays the **Specify Job Request Variables** panel.

HIST displays the **Display Job Request History** panel. This command is valid only when this panel was called from the **Modify Job Request** panel.

END returns to the **Add Job Request** panel or the **Modify Job Request** panel.

CANCEL cancels the add or modify and returns to the **Job Request Menu** panel, or the **List Job Requests for Owner** panel, without updating PJS.

UP *lines*	scrolls the display up by the specified number of lines.
DOWN *lines*	scrolls the display down by the specified number of lines.

Specify Job Request Variables Panel

The **Specify Job Request Variables** panel is used to enter and update job request variables The list of job request variables can be scrolled up and down to display more job request variables.

The **Specify Job Request Variables** panel is accessed from the **Add Job Request** panel or the **Modify Job Request** panel by entering the **VAR** command on the command line.

```
yy/mm/dd hh:mm -------  PJS - JOB REQUEST VARIABLES  ------- LINE nnnn OF nnnn
COMMAND ===> ____________________________________________________ SCROLL ===> scrl

Request-ID   ===> ownerid.reqname      Status       ===> status
Description  ===> description

    Target   Search   Shift    Repl
S   String  Strt End Type Col  Type  Replacement Value      Col nn to nn of nn
_  target__  nn  nn  type  nn  type  replacement value_____________________
_  xxxxxxxx  nn  nn  xxxx  nn  xxxx  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
_  xxxxxxxx  nn  nn  xxxx  nn  xxxx  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
_  xxxxxxxx  nn  nn  xxxx  nn  xxxx  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
_  xxxxxxxx  nn  nn  xxxx  nn  xxxx  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
_  ________  __  __  ____  __  ____  ________________________________________
_  ________  __  __  ____  __  ____  ________________________________________
_  ________  __  __  ____  __  ____  ________________________________________
_  ________  __  __  ____  __  ____  ________________________________________

Commands ==> JCL - JCL Source, FREQ - Frequency, EVENT - Events,
             OPT - Submit Options, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
Line Commands ==> I - Insert Variable, D - Delete Variable, R - Repeat Variable
```

Figure 44: Specify Job Request Variables Panel

The following fields are displayed, but may not be updated:

Request-ID
is the **Request-ID** of the job request being modified.

Status
is the **Job Request Status** of the job request. For a list of the possible values, see the section **The Job Request Status** on page 4.

Description
is the **Description** of the job request.

The following line commands may be entered in the selection field (under the **'S'** on the left side of the panel) for a job request event:

I inserts a blank entry for a new job request variable immediately above the selected line.

D deletes the job request variable.

R repeat the job request variable.

For each job request variable, the following data fields may be updated:

Target String
specifies the **Target String** of the job request variable. The target string is the JCL string to be replaced. This may be any character string. This parameter is required.

Search Strt
specifies the **Search Start Column** of the job request variable. The search for the target string begins in the specified column. This must be a number between 1 and 80. The default is 1.

Search End
specifies the **Search End Column** of the job request variable. The search for the target string ends in the specified column. This must be a number between 1 and 80, and must be greater than or equal to the **Search Start Column**. The default is 80.

Shift Type
specifies the Shift Type for the job request variable. Thew Shift Type specifies how data to the right of the target string is to be shifted when the length of the replacement string is not the same as the length of the target string. For more detailed information on how shifting works see the section **Specifying the Data Shift Options** on page 14. The Shift Type may be one of the following:

NONE do not shift

ALL shift all characters. This is the default.

SPACE shift characters into/from double spaces.

Shift Col
specifies the **Shift End Column** of the job request variable. Data will not be shifted or replaced past the specified column. This must be a number between 1 and 80, and must be greater than or equal to the **Search End Column**. The default is 80.

Repl Type

specifies the **type** of the **Replacement Value** for the job request variable. This may be one of the following:

LIT The replacement value is a **Literal Value**.

DYN The replacement value is a **Dynamic Value**.

VAR The replacement value is a **Global Variable**.

Replacement Value

specifies the **Replacement Value** of the job request variable. The format of the replacement value depends on the **Replacement Value Type**.

When the Replacement Value Type is **LIT** (Literal) this is a literal string. If quotes are not entered they will be automatically added. If the length of the value is too long to be displayed, the value may be scrolled left and right using the normal ISPF scrolling commands.

When the Replacement Value Type is **DYN** (Dynamic Value), this is the 1 to 16 character name of the dynamic value to be used. Dynamic values may be defined by PJS, or determined by an installation provided routine. The dynamic values provided by PJS are described in the section **Using Dynamic Values** on page 14. For a list of any installation provided routines, see your PJS System Administrator.

When the Replacement Value Type is **VAR** (Global Variable), this is the variable-ID of the global variable whose value is to be used as the replacement value. The format of a global variable-ID is described in the section **The Global Variable-ID** on page 25. Global variables are described in the section **Using Global Variables** on page 13.

The following commands may be entered on the **COMMAND** line:

JCL displays the **Specify Job Request JCL Source** panel.

FREQ displays the **Specify Job Request Frequency** panel.

EVENT displays the **Specify Job Request Events** panel.

OPT displays the **Specify Job Request Submit Options** panel.

HIST displays the **Display Job Request History** panel. This command is valid only when this panel was called from the **Modify Job Request** panel.

END returns to the **Add Job Request** panel or the **Modify Job Request** panel.

CANCEL cancels the add or modify and returns to the **Job Request Menu** panel, or the **List Job Requests for Owner** panel, without updating PJS.

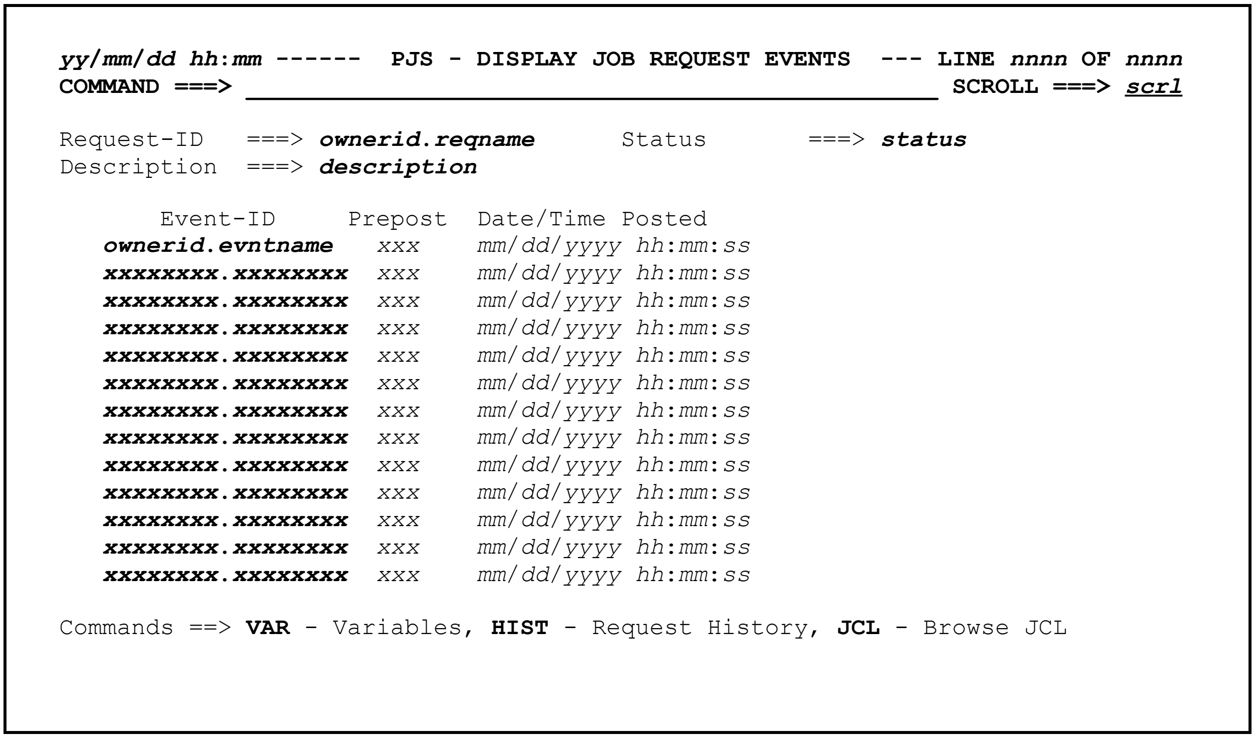
UP *lines* scrolls the display up by the specified number of lines.

DOWN *lines*	scrolls the display down by the specified number of lines.
LEFT *cols*	scrolls the literal replacement values left by the specified number of columns.
RIGHT *cols*	scrolls the literal replacement values right by the specified number of columns.

Display Job Request Events Panel

The **Display Job Request Events** panel is used to display job request events. The list of job request events can be scrolled up and down to display more job request events.

The **Display Job Request Events** panel is accessed from the **Delete Job Request** panel or the **Display Job Request** panel by entering the **EVENT** command on the command line.



```
yy/mm/dd hh:mm ------  PJS - DISPLAY JOB REQUEST EVENTS  --- LINE nnnn OF nnnn
COMMAND ===> ________________________________________________ SCROLL ===> scrl

Request-ID   ===> ownerid.reqname      Status       ===> status
Description  ===> description

       Event-ID     Prepost  Date/Time Posted
   ownerid.evntname   xxx    mm/dd/yyyy hh:mm:ss
   xxxxxxxx.xxxxxxxx  xxx    mm/dd/yyyy hh:mm:ss
   xxxxxxxx.xxxxxxxx  xxx    mm/dd/yyyy hh:mm:ss
   xxxxxxxx.xxxxxxxx  xxx    mm/dd/yyyy hh:mm:ss
   xxxxxxxx.xxxxxxxx  xxx    mm/dd/yyyy hh:mm:ss
   xxxxxxxx.xxxxxxxx  xxx    mm/dd/yyyy hh:mm:ss
   xxxxxxxx.xxxxxxxx  xxx    mm/dd/yyyy hh:mm:ss
   xxxxxxxx.xxxxxxxx  xxx    mm/dd/yyyy hh:mm:ss
   xxxxxxxx.xxxxxxxx  xxx    mm/dd/yyyy hh:mm:ss
   xxxxxxxx.xxxxxxxx  xxx    mm/dd/yyyy hh:mm:ss
   xxxxxxxx.xxxxxxxx  xxx    mm/dd/yyyy hh:mm:ss
   xxxxxxxx.xxxxxxxx  xxx    mm/dd/yyyy hh:mm:ss
   xxxxxxxx.xxxxxxxx  xxx    mm/dd/yyyy hh:mm:ss

Commands ==> VAR - Variables, HIST - Request History, JCL - Browse JCL
```

Figure 45: Display Job Request Events Panel

The following fields are displayed:

Request-ID
is the **Request-ID** of the job request being modified.

Status
is the **Job Request Status** of the job request. For a list of the possible values, see the section **The Job Request Status** on page 4.

Description
is the **Description** of the job request.

For each job request event, the following fields are displayed:

Event-ID
is the **Event-ID** of the job request event.

Prepost
specifies if the job request event may be posted before the **Next Run Date and Time** for the job request. **YES** means that the job request event can ne posted at any time. **NO** means that the job request evetn can only be posted if the **Next Run Date and Time** is past.

Date/Time Posted
If the job request event is currently posted, this is the date and time the job request event was posted. If the job request event is not currently posted, this field is blank.

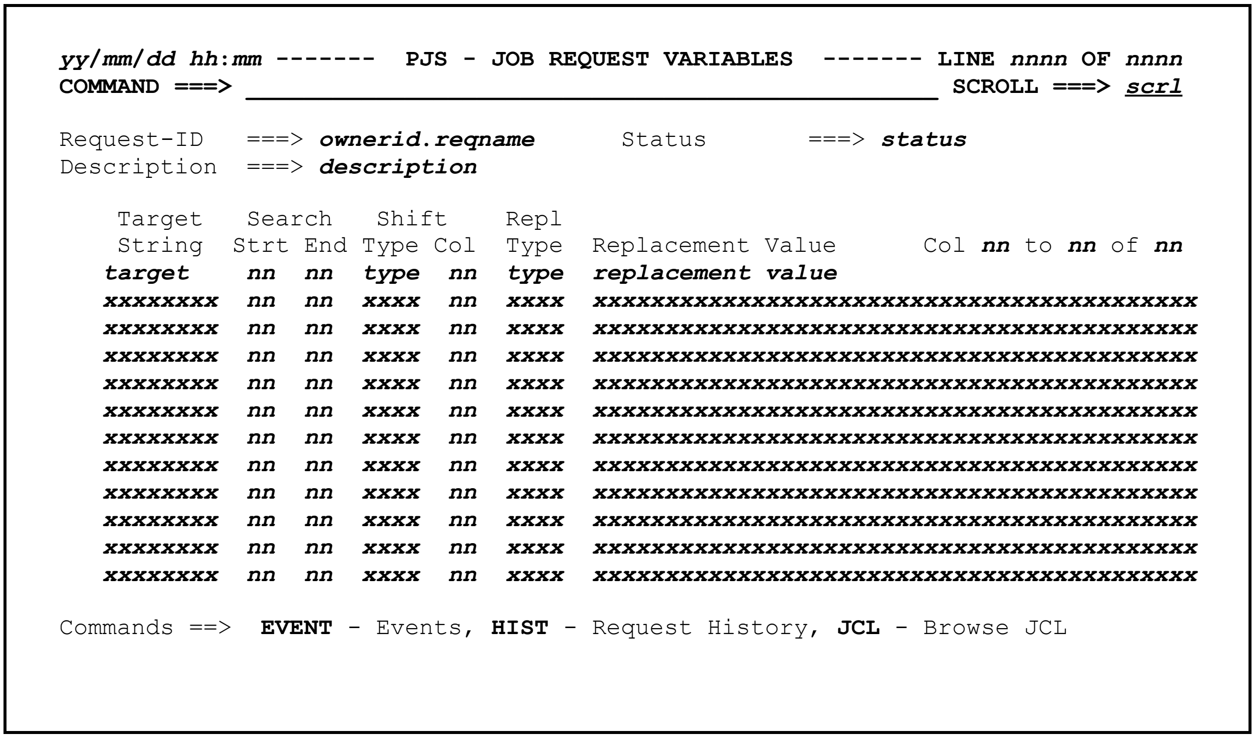
The following commands may be entered on the **COMMAND** line:

VAR	displays the **Display Job Request Variables** panel.
HIST	displays the **Display Job Request History** panel. This command is valid only when this panel was called from the **Modify Job Request** panel.
JCL	displays the JCL to be submitted, either from the user's JCL data set, or the PJS JCL Spool, using ISPF Browse.
END	returns to the **Delete Job Request** panel or the **Display Job Request** panel.
UP *lines*	scrolls the display up by the specified number of lines.
DOWN *lines*	scrolls the display down by the specified number of lines.

Display Job Request Variables Panel

The **Display Job Request Variables** panel is used to display job request variables. The list of job request variables can be scrolled up and down to display more job request variables.

The **Display Job Request Variables** panel is accessed from the **Delete Job Request** panel or the **Display Job Request** panel by entering the **VAR** command on the command line.



```
yy/mm/dd hh:mm -------  PJS - JOB REQUEST VARIABLES  ------- LINE nnnn OF nnnn
COMMAND ===> ________________________________________________ SCROLL ===> scrl

Request-ID   ===> ownerid.reqname      Status       ===> status
Description  ===> description

    Target   Search   Shift    Repl
    String  Strt End Type Col  Type  Replacement Value      Col nn to nn of nn
   target    nn  nn  type  nn  type  replacement value
   xxxxxxxx  nn  nn  xxxx  nn  xxxx  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
   xxxxxxxx  nn  nn  xxxx  nn  xxxx  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
   xxxxxxxx  nn  nn  xxxx  nn  xxxx  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
   xxxxxxxx  nn  nn  xxxx  nn  xxxx  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
   xxxxxxxx  nn  nn  xxxx  nn  xxxx  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
   xxxxxxxx  nn  nn  xxxx  nn  xxxx  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
   xxxxxxxx  nn  nn  xxxx  nn  xxxx  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
   xxxxxxxx  nn  nn  xxxx  nn  xxxx  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
   xxxxxxxx  nn  nn  xxxx  nn  xxxx  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
   xxxxxxxx  nn  nn  xxxx  nn  xxxx  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
   xxxxxxxx  nn  nn  xxxx  nn  xxxx  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

Commands ==>  EVENT - Events, HIST - Request History, JCL - Browse JCL
```

Figure 46: Display Job Request Variables Panel

The following fields are displayed:

Request-ID
is the **Request-ID** of the job request being modified.

Status
is the **Job Request Status** of the job request. For a list of the possible values, see the section **The Job Request Status** on page 4.

Description
is the **Description** of the job request.

For each job request variable, the following fields are displayed:

Target String
is the **Target String** of the job request variable. The target string is the JCL string to be replaced.

Search Strt
is the **Search Start Column** of the job request variable. The search for the target string begins in the specified column.

Search End
is the **Search End Column** of the job request variable. The search for the target string ends in the specified column.

Shift Type
is the Shift Type for the job request variable. Thew Shift Type specifies how data to the right of the target string is to be shifted when the length of the replacement string is not the same as the length of the target string. The Shift Type may be one of the following:

NONE do not shift

ALL shift all characters.

SPACE shift characters into/from double spaces.

Shift Col
is the **Shift End Column** of the job request variable. Data will not be shifted or replaced past the specified column.

Repl Type
is the **type** of the **Replacement Value** for the job request variable. This may be one of the following:

LIT The replacement value is a **Literal Value**.

DYN The replacement value is a **Dynamic Value**.

VAR The replacement value is a **Global Variable**.

Replacement Value
is the **Replacement Value** of the job request variable. The format of the replacement value depends on the **Replacement Value Type**.

When the Replacement Value Type is **LIT** (Literal) this is a literal string. If the length of the value is too long to be displayed, the value may be scrolled left and right using the normal ISPF scrolling commands.

When the Replacement Value Type is **DYN** (Dynamic Value), this is the name of the dynamic value to be used. Dynamic values may be defined by PJS, or determined by an installation provided routine. The dynamic values provided by PJS are described in the section **Using Dynamic Values** on page 14. For a list of any installation provided routines, see your PJS System Administrator.

When the Replacement Value Type is **VAR** (Global Variable), this is the variable-ID of the global variable whose value is to be used as the replacement value. Global variables are described in the section **Using Global Variables** on page 13.

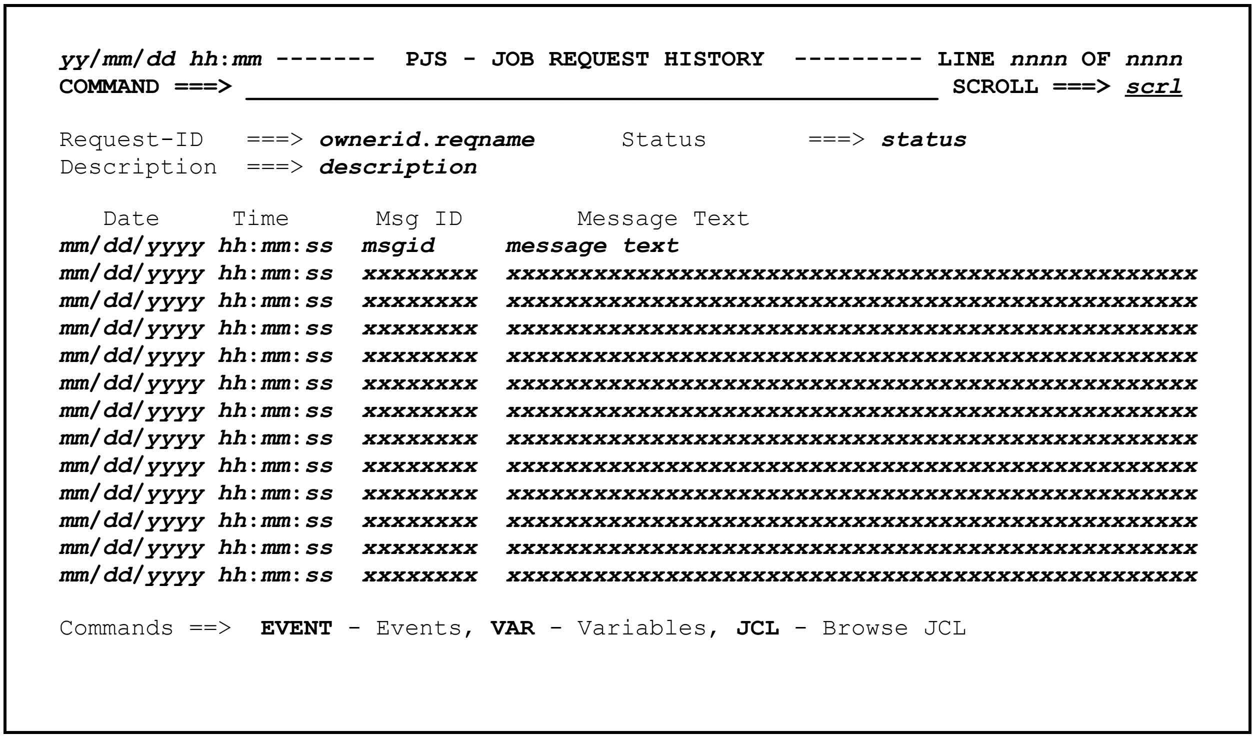
The following commands may be entered on the **COMMAND** line:

EVENT	displays the **Display Job Request Events** panel.
HIST	displays the **Display Job Request History** panel. This command is valid only when this panel was called from the **Modify Job Request** panel.
JCL	displays the JCL to be submitted, either from the user's JCL data set, or the PJS JCL Spool, using ISPF Browse.
END	returns to the **Delete Job Request** panel or the **Display Job Request** panel.
UP *lines*	scrolls the display up by the specified number of lines.
DOWN *lines*	scrolls the display down by the specified number of lines.
LEFT *cols*	scrolls the literal replacement value left by the specified number of columns.
RIGHT *cols*	scrolls the literal replacement value right by the specified number of columns.

Display Job Request History Panel

The **Display Job Request History** panel is used to display the message history for a job request. The PJS Message History Log contains most messages sent to the user by the PJS System Task. The list of messages can be scrolled up and down to display more messages, and left and right to display additional text for each message.

The **Display Job Request History** panel is accessed from the **Modify Job Request** panel, the **Delete Job Request** panel, or the **Display Job Request** panel by entering the **HIST** command on the command line.



```
yy/mm/dd hh:mm -------  PJS - JOB REQUEST HISTORY  --------- LINE nnnn OF nnnn
COMMAND ===> ________________________________________________ SCROLL ===> scrl

Request-ID   ===> ownerid.reqname      Status       ===> status
Description  ===> description

   Date     Time      Msg ID        Message Text
mm/dd/yyyy hh:mm:ss  msgid     message text
mm/dd/yyyy hh:mm:ss  xxxxxxxx  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
mm/dd/yyyy hh:mm:ss  xxxxxxxx  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
mm/dd/yyyy hh:mm:ss  xxxxxxxx  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
mm/dd/yyyy hh:mm:ss  xxxxxxxx  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
mm/dd/yyyy hh:mm:ss  xxxxxxxx  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
mm/dd/yyyy hh:mm:ss  xxxxxxxx  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
mm/dd/yyyy hh:mm:ss  xxxxxxxx  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
mm/dd/yyyy hh:mm:ss  xxxxxxxx  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
mm/dd/yyyy hh:mm:ss  xxxxxxxx  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
mm/dd/yyyy hh:mm:ss  xxxxxxxx  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
mm/dd/yyyy hh:mm:ss  xxxxxxxx  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
mm/dd/yyyy hh:mm:ss  xxxxxxxx  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

Commands ==>  EVENT - Events, VAR - Variables, JCL - Browse JCL
```

Figure 47: Display Job Request History Panel

The following fields are displayed:

Request-ID
is the **Request-ID** of the job request being modified.

Status
is the **Job Request Status** of the job request. For a list of the possible values, see the section **The Job Request Status** on page 4.

Description
is the **Description** of the job request.

For each message, the following fields are displayed:

Date
is the date the message was issued.

Time
is the time the message was issued.

Msg Id
is the **Message ID** of the message.

Message Text
is the text of the message

If the **Display Job Request History** panel was entered from the **Modify Job Request** panel, the following commands may be entered on the **COMMAND** line:

JCL	displays the **Specify Job Request JCL Source** panel.
FREQ	displays the **Specify Job Request Frequency** panel.
EVENT	displays the **Specify Job Request Events** panel.
OPT	displays the **Specify Job Request Submit Options** panel.
VAR	displays the **Specify Job Request Variables** panel.
CANCEL	cancels the modify and returns to the **Job Request Menu** panel, or the **List Job Requests for Owner** panel, without updating PJS.

If the **Display Job Request History** panel was entered from the **Delete Job Request** panel or the **Display Job Request** panel, the following commands may be entered on the **COMMAND** line:

EVENT	displays the **Display Job Request Events** panel.
VAR	displays the **Display Job Request Variables** panel.
JCL	displays the JCL to be submitted, either from the user's JCL data set, or the PJS JCL Spool, using ISPF Browse.

In either case, the following commands may be entered on the **COMMAND** line:

END	returns to the **Modify Job Request** panel, the **Delete Job Request** panel, or the **Display Job Request** panel.
UP *lines*	scrolls the display up by the specified number of lines.
DOWN *lines*	scrolls the display down by the specified number of lines.
LEFT *cols*	scrolls the message text left by the specified number of columns.
RIGHT *cols*	scrolls the message text right by the specified number of columns.

Display Job Request Internal Information Panel

The **Display Job Request Internal Information** panel is used to display information about a job request that is not normally of concern to the user, but which may be useful in diagnosing problems with PJS.

The **Display Job Request Internal Information** panel is accessed from the **Add Job Request** panel, the **Modify Job Request** panel, the **Delete Job Request** panel, or the **Display Job Request** panel by entering the **INT** command on the command line.

```
yy/mm/dd hh:mm -------  PJS - DISPLAY INTERNAL INFO  -------------------------
COMMAND ===> ____________________________________________________________________

Request-ID   ===> ownerid.reqname      Status       ===> status
Description  ===> description

Status Changed   ===> mm/dd/yyyy hh:mm:ss

Spool Member     ===> member
Spool Records    ===> nnnnnnnnnn

Submit Retries   ===> nnnnn


                  installation data heading
Inst Data    ===> installation data



Commands ==> EVENT - Events, VAR - Variables, HIST - Request History,
             JCL - Browse JCL
```

Figure 48: Display Job Request Internal Information Panel

The following fields are displayed:

Request-ID
is the **Request-ID** of the job request being modified.

Status
is the **Job Request Status** of the job request. For a list of the possible values, see the section **The Job Request Status** on page 4.

Description
is the **Description** of the job request.

Status Changed D/T
is the **Last Status Change Date and Time** of the job request. This is this date and time at which the Job Request Status last changed. If changes to the job request have been made that will change the Job Request Status, but the update has not yet been completed, this field will indicate **PENDING**.

Spool Member
is the PJS JCL Spool member name in which the user's JCL has been saved. If the JCL has not been saved, this field is blank.

Spool Records
is the number of JCL records saved in the PJS JCL Spool for this job request.

Submit Retry
is the number of times the PJS System Task has attempted to submit the job, but was unable to because of a temporay error (usually because the user's JCL data set is in use by another task).

Inst Data
is installation defined data. This is usually used for security related information, and may not be present. For detailed information on your installation data, see your PJS System Administrator.

If the **Display Job Request History** panel was entered from the **Add Job Request** panel or the **Modify Job Request** panel, the following commands may be entered on the **COMMAND** line:

JCL displays the **Specify Job Request JCL Source** panel.

FREQ displays the **Specify Job Request Frequency** panel.

EVENT displays the **Specify Job Request Events** panel.

OPT displays the **Specify Job Request Submit Options** panel.

VAR displays the **Specify Job Request Variables** panel.

CANCEL cancels the add or modify and returns to the **Job Request Menu** panel, or the **List Job Requests for Owner** panel, without updating PJS.

If the **Display Job Request History** panel was entered from the **Delete Job Request** panel or the **Display Job Request** panel, the following commands may be entered on the **COMMAND** line:

EVENT displays the **Display Job Request Events** panel.

VAR displays the **Display Job Request Variables** panel.

JCL displays the JCL to be submitted, either from the user's JCL data set, or the PJS JCL Spool, using ISPF Browse.

In either case, the following commands may be entered on the **COMMAND** line:

HIST displays the **Display Job Request History** panel.

END returns to the **Add Job Request** panel, the **Modify Job Request** panel, the **Delete Job Request** panel, or the **Display Job Request** panel.

Calendar Panels

The Calendar panels allow you to add, modify, delete, and display PJS Calendars.

Calendar Menu Panel

The **Calendar Menu** panel provides access to the calendar panels. Inputs are provided for entering the Owner-ID and Calendar Name of the calendar to be updated or displayed.

The **Calendar Menu** panel is accessed from the **PJS Main Menu** panel by selecting option **C**.

```
yy/mm/dd hh:mm -------  PJS - CALENDAR MENU  ----------------------------------
OPTION ===> ____________________________________________________________________

Select one of the following functions:

  L  LIST     - List all Calendars for Owner-ID
  A  ADD      - Add a new Calendar
  D  DELETE   - Delete a Calendar
  M  MODIFY   - Modify a Calendar
  S  DISPLAY  - Display a Calendar


Calendar-ID:
  Owner-ID      ===> ownerid_
  Calendar Name ===> calname_       (Required for options A, D, M, and S)


Enter Calendar-ID (if required) and make a selection, then press ENTER.
```

Figure 49: PJS Calendar Menu Panel

To use this panel, enter the owner-ID and calendar name of the calendar to be updated or displayed. Then, enter the letter corresponding to the option desired in the **OPTION** field and press **ENTER**. To return to the PJS Main Menu panel, enter **END** on the option line (or press the **END** PF-key, usually **PF3**).

The following options may be entered on the **OPTION** line:

A selects the **Add Calendar** panel.

D selects the **Delete Calendar** panel.

L selects the **List Calendars** for Owner panel.

M selects the **Modify Calendar** panel.

S selects the **Display Calendar** panel.

END returns to the **PJS Main Menu** panel.

The following record key fields may be entered:

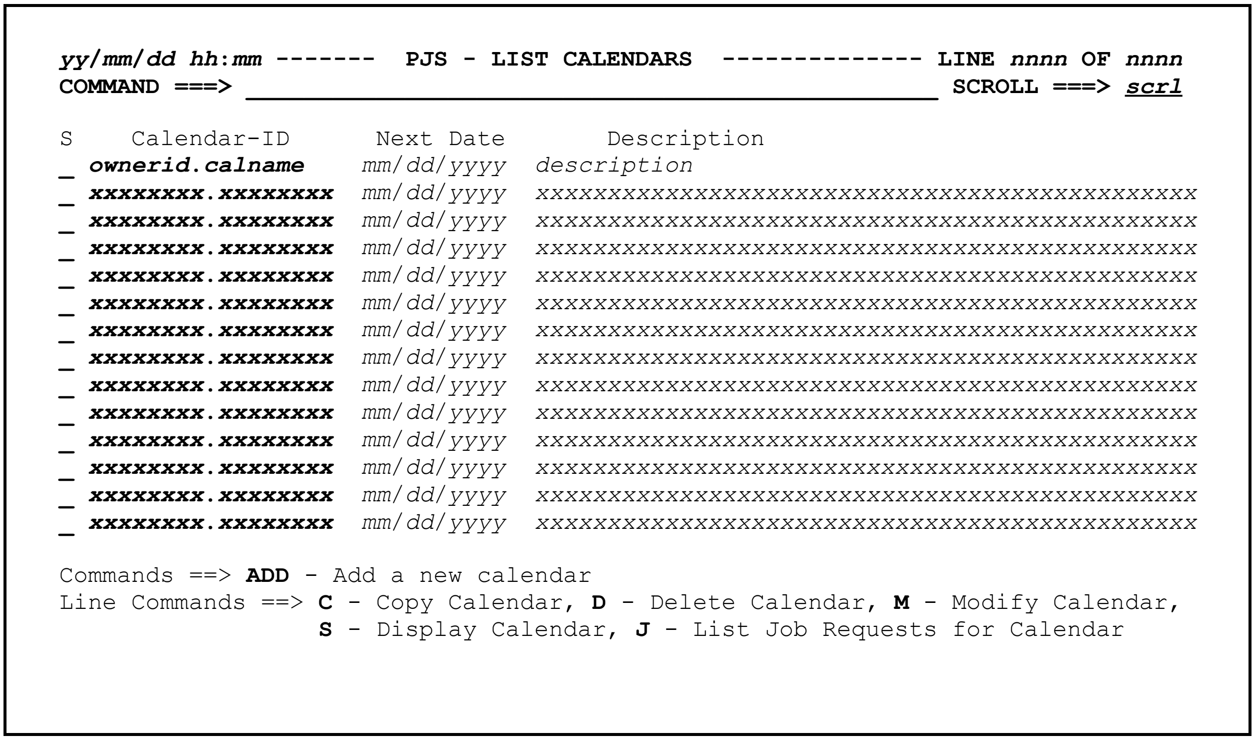
Owner-ID
specifies the **Owner-ID** of the calendar(s) to be updated or displayed. This field is required.

Calendar Name
specifies the **Calendar Name** of the calendar to be updated or displayed. The format of the calendar name is described in the section **The Calendar-ID** on page 24. This field is required when option **A** (Add Calendar), option **D** (Delete Calendar), **M** (Modify Calendar), or **S** (Display Calendar) is selected. It is ignored when option **L** (List Calendars for Owner) is selected.

List Calendars for Owner Panel

The **List Calendars for Owner** panel lists the calendars owned by the current Owner-ID. The list can be scrolled up and down to display more calendars, and left and right to display additional information for each calendar. Individual calendars may then be selected for update or display using line commands.

The **List Calendars for Owner** panel is accessed from the **Calendar Menu** panel by selecting option **L**.



```
yy/mm/dd hh:mm -------  PJS - LIST CALENDARS  -------------- LINE nnnn OF nnnn
COMMAND ===> ________________________________________________ SCROLL ===> scrl

S    Calendar-ID      Next Date       Description
_ ownerid.calname    mm/dd/yyyy  description
_ xxxxxxxx.xxxxxxxx  mm/dd/yyyy  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
_ xxxxxxxx.xxxxxxxx  mm/dd/yyyy  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
_ xxxxxxxx.xxxxxxxx  mm/dd/yyyy  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
_ xxxxxxxx.xxxxxxxx  mm/dd/yyyy  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
_ xxxxxxxx.xxxxxxxx  mm/dd/yyyy  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
_ xxxxxxxx.xxxxxxxx  mm/dd/yyyy  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
_ xxxxxxxx.xxxxxxxx  mm/dd/yyyy  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
_ xxxxxxxx.xxxxxxxx  mm/dd/yyyy  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
_ xxxxxxxx.xxxxxxxx  mm/dd/yyyy  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
_ xxxxxxxx.xxxxxxxx  mm/dd/yyyy  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
_ xxxxxxxx.xxxxxxxx  mm/dd/yyyy  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
_ xxxxxxxx.xxxxxxxx  mm/dd/yyyy  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
_ xxxxxxxx.xxxxxxxx  mm/dd/yyyy  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

Commands ==> ADD - Add a new calendar
Line Commands ==> C - Copy Calendar, D - Delete Calendar, M - Modify Calendar,
                  S - Display Calendar, J - List Job Requests for Calendar
```

Figure 50: List Calendars for Owner Panel

For each calendar, the following fields are displayed:

Calendar-ID
is the **Calendar-ID** of the calendar.

Next Date
is the next date selected by the calendar.

Description
is the **Description** of the calendar.

The following line commands may be entered in the selection field (under the **'S'** on the left side of the panel) for a calendar:

C selects the **Add Calendar** panel to add a new calendar using the selected calendar as a model.

D selects the **Delete Calendar** panel for the selected calendar.

M selects the **Modify Calendar** panel for the selected calendar.

S selects the **Display Calendar** panel for the selected calendar.

J selects the **Display Job Requests for Calendar** panel for the selected calendar.

The following commands may be entered on the **COMMAND** line:

ADD selects the **Add Calendar** panel for a new calendar.

END returns to the **Calendar Menu** panel.

UP *lines* scrolls the display up by the specified number of lines.

DOWN *lines* scrolls the display down by the specified number of lines.

LEFT *cols* scrolls the display left by the specified number of columns.

RIGHT *cols* scrolls the display right by the specified number of columns.

Add Calendar Panel

The **Add Calendar** panel is used to add a new calendar. When this panel is first displayed, a skeleton calendar containing only default information is created in the ISPF user's storage. The user then selects the desired dates, scrolling through the calendar as necessary. The add is not completed until the **END** command is entered, indicating that all dates have been selected. At any time prior to entering the END command, the user may cancel the add by entering the **CANCEL** command.

The calendar dates are displayed, a month at a time, in a typical calendar format. The month displayed can be selected by entering the month and year desired in the fields provided. You can also 'scroll' through the calendar, from one month to the next, by using the ISPF **UP** and **DOWN** scroll commands (or their corresponding PF keys).

The **Add Calendar** panel is accessed from the **Calendar Menu** panel by selecting option **A**, or from the **List Calendars for Owner** panel by either entering the **ADD** command on the command line, or by entering a **C** line command on an existing calendar.

If the **Add Calendar** dialog is called from the **List Calendars for Owner** panel, the **Add Calendar-ID** panel will be displayed first, to prompt the user for the Calendar-ID to be added.

```
yy/mm/dd hh:mm -------  PJS - ADD CALENDAR  -----------------------------------
COMMAND ===> ____________________________________________________________________

Calendar-ID  ===> ownerid.calname
Description  ===> description_____________________________________

                Month month____                Year year

                Sun   Mon   Tue   Wed   Thu   Fri   Sat
               _ dd  _ dd  _ dd  _ dd  _ dd  _ dd  _ dd
               _ dd  _ dd  _ dd  _ dd  _ dd  _ dd  _ dd
               _ dd  _ dd  _ dd  _ dd  _ dd  _ dd  _ dd
               _ dd  _ dd  _ dd  _ dd  _ dd  _ dd  _ dd
               _ dd  _ dd  _ dd  _ dd  _ dd  _ dd  _ dd
               _ dd  _ dd

Enter the 3-letter month and year above to select month, or
Enter scroll commands (or use PF keys) to select next/previous month

Enter a non-blank character before each date to be selected
Enter a Space before each date to be excluded

Commands ==> REQ - Display Job Requests for Calendar,
             END - Complete Add, CANCEL - Cancel Add
```

Figure 51: Add Calendar Panel

The following fields are displayed, but may not be updated:

Calendar-ID
is the **Calendar-ID** of the calendar being added. This parameter cannot be updated.

The following data fields may be updated

Description
specifies the **Description** of the calendar. This is for documentation purposes only and does not affect PJS processing in any way.

The following month display selection fields may be updated

Month
specifies the name of the calendar month to be displayed. Only the first three letters of the month name are required.

Year
specifies the 4-digit calendar year to be displayed.

To select a date, first display the calendar month of the date, then enter any non-blank character in the unprotected field immediately before the date. To omit a date, enter a space in the unprotected field before the date. The dates selected will be displayed in high intensity and an asterisk ('*') is inserted in the unprotected field before the date. Dates not selected will be displayed in low intensity.

The following commands may be entered on the **COMMAND** line:

REQ	displays the **Display Job Requests For Calendar** panel.
END	completes the add and returns to the **Calendar Menu** panel, or the **List Calendars for Owner** panel.
CANCEL	cancels the add and returns to the **Calendar Menu** panel, or the **List Calendars for Owner** panel, without updating PJS.
UP *months*	scrolls the calendar backward by the specified number of months.
DOWN *months*	scrolls the calendar forward by the specified number of months.

Add Calendar-ID Panel

The **Add Calendar-ID** panel is used to specify the Calendar-ID when the **Add Calendar** dialog is entered from the **List Calendars for Owner** panel. This panel is displayed before the **Add Calendar** panel. After this panel is completed, the **Add Calendar** panel will be displayed.

```
yy/mm/dd hh:mm -------  PJS - ADD CALENDAR ID  --------------------------------
COMMAND ===> ____________________________________________________________________

New Calendar-ID:
  Owner-ID      ===> ownerid_
  Calendar Name ===> calname_

Enter New Calendar-ID, then press ENTER.









Commands ==> CANCEL - Cancel Add
```

Figure 52: Add Calendar-ID Panel

The following record key fields may be entered

Owner-ID
specifies the **Owner-ID** of the calendar to be added. This field is required.

Calendar Name
specifies the **Calendar Name** of the calendar to be added. The format of the calendar name is described in the section **The Calendar-ID** on page 24. This field is required.

The following commands may be entered on the **COMMAND** line:

END returns to the **List Calendars for Owner** panel.

CANCEL returns to the **List Calendars for Owner** panel.

Modify Calendar Panel

The **Modify Calendar** panel is used to modify an existing calendar. When this panel is first displayed, the calendar is read from the PJS Request Queue and saved in the ISPF user's storage. The user then selects the desired dates, scrolling through the calendar as necessary. The modify is not completed until the **END** command is entered, indicating that all dates have been selected. At any time prior to entering the END command, the user may cancel the modify by entering the **CANCEL** command.

The calendar dates are displayed, a month at a time, in a typical calendar format. The month displayed can be selected by entering the month and year desired in the fields provided. You can also 'scroll' through the calendar, from one month to the next, by using the ISPF **UP** and **DOWN** scroll commands (or their corresponding PF keys).

When changing a calendar, the user should be aware that any job requests that specify the calendar will **not** have the **Next Run Date and Time** updated to reflect the changes to the calendar until after the next time the job request is submitted by the PJS System Task. If the Next Run Date and Time needs to be updated immediately, it is necessary to use the **PJREQMOD** TSO command (using the **RESET RUNTIME** sub-command) or the **Modify Job Request** ISPF dialog (specifying **YES** in the **Recalculate** field of the **Specify Job Request Frequency** panel) to recalculate the Next Run Date and Time. To determine which job requests use a calendar, use the **List Job Requests for Calendar** ISPF dialog.

The **Modify Calendar** panel is accessed from the **Calendar Menu** panel by selecting option **M**, or from the **List Calendars for Owner** panel by entering a **C** line command on an existing calendar.

```
yy/mm/dd hh:mm -------  PJS - MODIFY CALENDAR  --------------------------------
COMMAND ===> ____________________________________________________________________

Calendar-ID  ===> ownerid.calname
Description  ===> description_____________________________________

               Month month____                Year year

               Sun   Mon   Tue   Wed   Thu   Fri   Sat
              _ dd  _ dd  _ dd  _ dd  _ dd  _ dd  _ dd
              _ dd  _ dd  _ dd  _ dd  _ dd  _ dd  _ dd
              _ dd  _ dd  _ dd  _ dd  _ dd  _ dd  _ dd
              _ dd  _ dd  _ dd  _ dd  _ dd  _ dd  _ dd
              _ dd  _ dd  _ dd  _ dd  _ dd  _ dd  _ dd
              _ dd  _ dd

Enter the 3-letter month and year above to select month, or
Enter scroll commands (or use PF keys) to select next/previous month

Enter a non-blank character before each date to be selected
Enter a Space before each date to be excluded

Commands ==> REQ - Display Job Requests for Calendar
             END - Complete Modify, CANCEL - Cancel Modify
```

Figure 53: Modify Calendar Panel

The following fields are displayed, but may not be updated:

Calendar-ID
is the **Calendar-ID** of the calendar being modified. This parameter cannot be updated.

The following data fields may be updated

Description
specifies the **Description** of the calendar. This is for documentation purposes only and does not affect PJS processing in any way.

The following month display selection fields may be updated

Month
specifies the name of the calendar month to be displayed. Only the first three letters of the month name are required.

Year
specifies the 4-digit calendar year to be displayed.

To select a date, first display the calendar month of the date, then enter any non-blank character in the unprotected field immediately before the date. To omit a date, enter a space in the unprotected field before the date. The dates selected will be displayed in high intensity and an asterisk ('*') is inserted in the unprotected field before the date. Dates not selected will be displayed in low intensity.

The following commands may be entered on the **COMMAND** line:

REQ	displays the **Display Job Requests For Calendar** panel.
END	completes the modify and returns to the **Calendar Menu** panel, or the **List Calendars for Owner** panel.
CANCEL	cancels the modify and returns to the **Calendar Menu** panel, or the **List Calendars for Owner** panel, without updating PJS.
UP *months*	scrolls the calendar backward by the specified number of months.
DOWN *months*	scrolls the calendar forward by the specified number of months.

Delete Calendar Panel

The **Delete Calendar** panel is used to delete a calendar. To confirm the delete, the user must enter **YES** in the confirmation field. The delete is then completed by entering the **END** command. The user may cancel the delete by entering the **CANCEL** command.

The calendar dates are displayed, a month at a time, in a typical calendar format. The month displayed can be selected by entering the month and year desired in the fields provided. You can also 'scroll' through the calendar, from one month to the next, by using the ISPF **UP** and **DOWN** scroll commands (or their corresponding PF keys).

The user should beware that it is possible to delete a calendar that is currently being used by a job request. If this is done, there will be no indication until the PJS System Task submits the job for the job request at its scheduled **Next Run Date and Time**. After the job is submitted, the reschedule for the job request will fail with message PJS2221E, and the job request will be placed in the ERROR status. To determine which job requests use a calendar, use the **List Job Requests for Calendar** ISPF dialog.

The **Delete Calendar** panel is accessed from the **Calendar Menu** panel by selecting option **D**, or from the **List Calendars for Owner** panel by entering a **D** line command on an existing calendar.

```
yy/mm/dd hh:mm -------  PJS - DELETE CALENDAR  --------------------------------
COMMAND ===> ____________________________________________________________________

Calendar-ID  ===> ownerid.calname
Description  ===> description

               Month month____                 Year year

               Sun   Mon   Tue   Wed   Thu   Fri   Sat
                dd    dd    dd    dd    dd    dd    dd
                dd    dd    dd    dd    dd    dd    dd
                dd    dd    dd    dd    dd    dd    dd
                dd    dd    dd    dd    dd    dd    dd
                dd    dd    dd    dd    dd    dd    dd
                dd    dd

Enter the 3-letter month and year above to select month, or
Enter scroll commands (or use PF keys) to select next/previous month

Enter 'YES' and press enter to confirm delete ===> ___

Commands ==> REQ - Display Job Requests for Calendar,
             END - Complete Delete, CANCEL - Cancel Delete
```

Figure 54: Delete Calendar Panel

The following fields are displayed, but may not be updated:

Calendar-ID
is the **Calendar-ID** of the calendar being modified. This parameter cannot be updated.

Description
specifies the **Description** of the calendar. This is for documentation purposes only and does not affect PJS processing in any way.

The following month display selection fields may be updated

Month
specifies the name of the calendar month to be displayed. Only the first three letters of the month name are required.

Year
specifies the 4-digit calendar year to be displayed.

The dates selected will be displayed in high intensity, and an asterisk ('*') appears immediatly before the date. Dates not selected will be displayed in low intensity.

The following commands may be entered on the **COMMAND** line:

REQ	displays the **Display Job Requests For Calendar** panel.
END	completes the delete and returns to the **Calendar Menu** panel, or the **List Calendars for Owner** panel.
CANCEL	cancels the delete and returns to the **Calendar Menu** panel, or the **List Calendars for Owner** panel, without updating PJS.
UP *months*	scrolls the calendar backward by the specified number of months.
DOWN *months*	scrolls the calendar forward by the specified number of months.

Display Calendar Panel

The **Display Calendar** panel is used to delete a calendar.

The calendar dates are displayed, a month at a time, in a typical calendar format. The month displayed can be selected by entering the month and year desired in the fields provided. You can also 'scroll' through the calendar, from one month to the next, by using the ISPF **UP** and **DOWN** scroll commands (or their corresponding PF keys).

The **Display Calendar** panel is accessed from the **Calendar Menu** panel by selecting option **S**, or from the **List Calendars for Owner** panel by entering an **S** line command on an existing calendar.

```
yy/mm/dd hh:mm -------  PJS - DISPLAY CALENDAR  -------------------------------
COMMAND ===> ______________________________________________________________________

Calendar-ID  ===> ownerid.calname
Description  ===> description

               Month month____                 Year year

               Sun   Mon   Tue   Wed   Thu   Fri   Sat
                dd    dd    dd    dd    dd    dd    dd
                dd    dd    dd    dd    dd    dd    dd
                dd    dd    dd    dd    dd    dd    dd
                dd    dd    dd    dd    dd    dd    dd
                dd    dd    dd    dd    dd    dd    dd
                dd    dd

Enter the 3-letter month and year above to select month, or
Enter scroll commands (or use PF keys) to select next/previous month


Commands ==> REQ - Display Job Requests for Calendar
```

Figure 55: Display Calendar Panel

The following fields are displayed:

Calendar-ID
is the **Calendar-ID** of the calendar being modified. This parameter cannot be updated.

Description
specifies the **Description** of the calendar. This is for documentation purposes only and does not affect PJS processing in any way.

The following month display selection fields may be updated

Month
specifies the name of the calendar month to be displayed. Only the first three letters of the month name are required.

Year
specifies the 4-digit calendar year to be displayed.

The dates selected will be displayed in high intensity, and an asterisk ('*') appears immediatly before the date. Dates not selected will be displayed in low intensity.

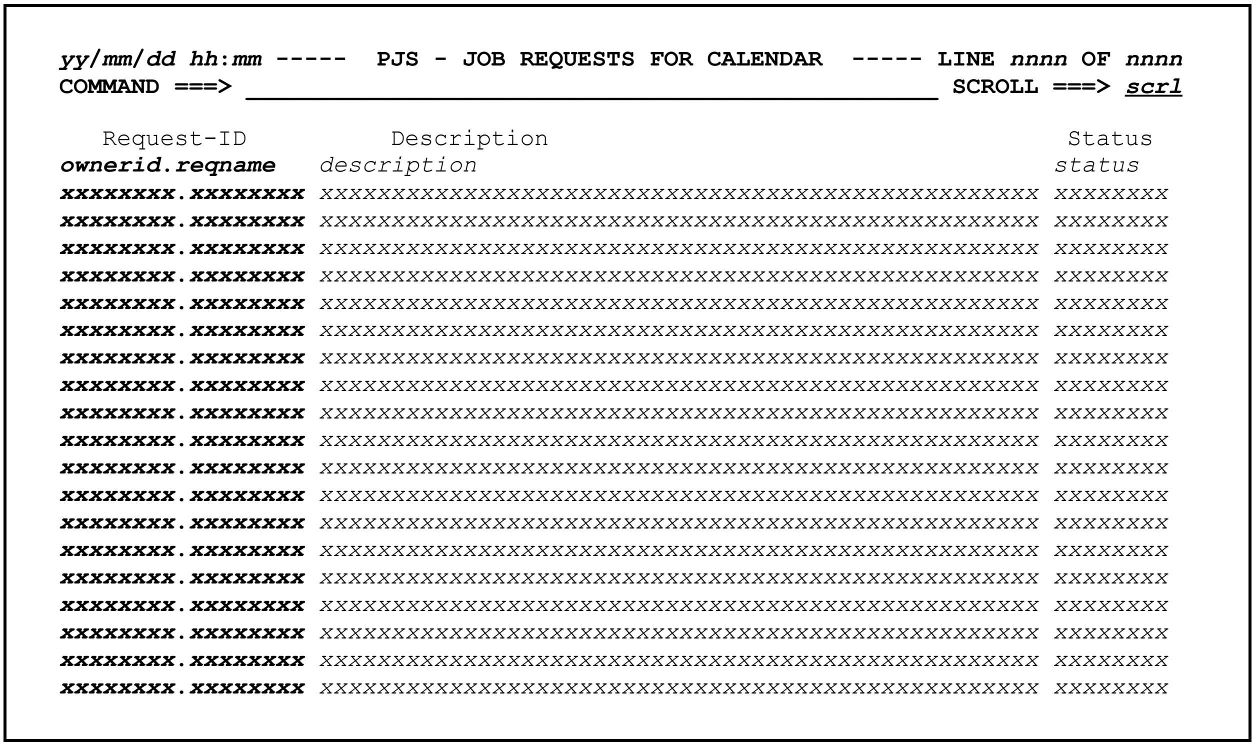
The following commands may be entered on the **COMMAND** line:

REQ	displays the **Display Job Requests For Calendar** panel.
END	returns to the **Calendar Menu** panel, or the **List Calendars for Owner** panel.
UP *months*	scrolls the calendar backward by the specified number of months.
DOWN *months*	scrolls the calendar forward by the specified number of months.

List Job Requests for Calendar Panel

The **List Job Requests for Calendar** panel lists the job requests that specify a calendar. The list can be scrolled up and down to display more job requests, and left and right to display additional information for each job request.

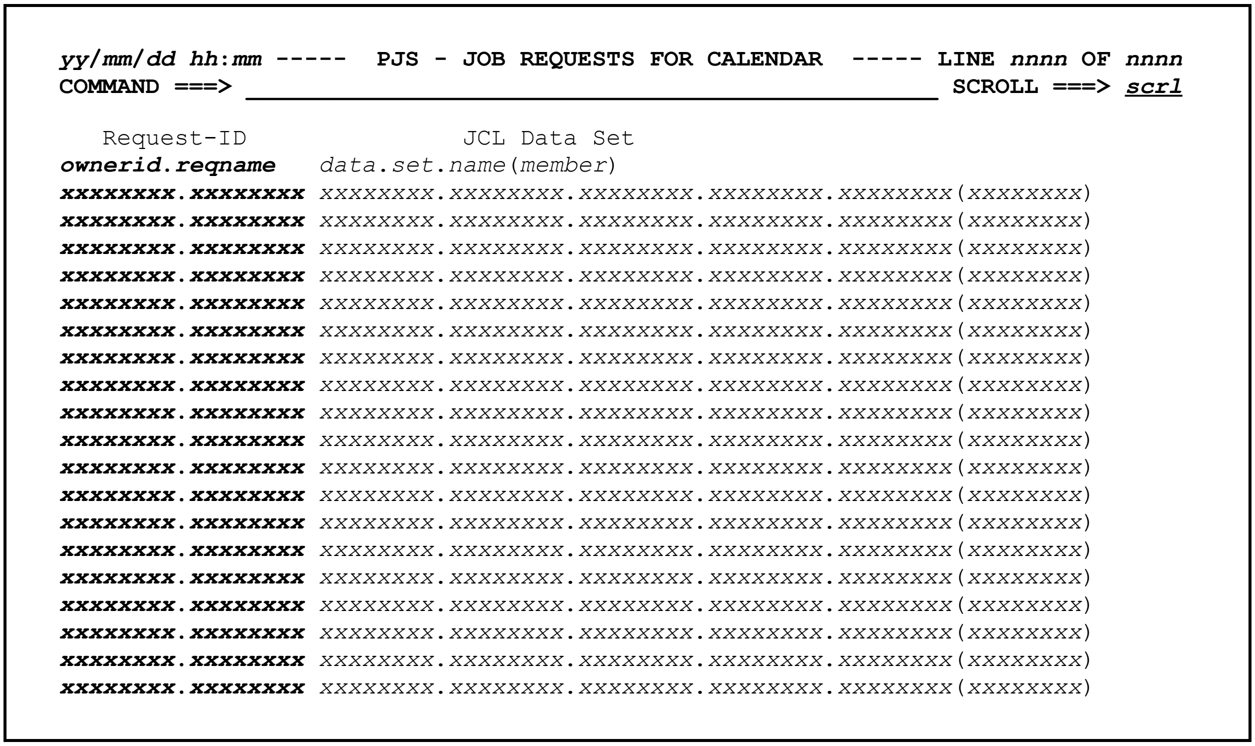
The **List Job Requests for Calendar** panel is accessed from the **Add Calendar** panel, the **Modify Calendar** panel, the **Delete Calendar** panel, or the **Display Calendar** panel, by entering the **REQ** command on the command line, or from the **List Calendars for Owner** panel by entering a **J** line command on an existing calendar.



```
yy/mm/dd hh:mm -----  PJS - JOB REQUESTS FOR CALENDAR  ----- LINE nnnn OF nnnn
COMMAND ===> ________________________________________________ SCROLL ===> scrl

   Request-ID         Description                                    Status
ownerid.reqname   description                                       status
xxxxxxxx.xxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx
xxxxxxxx.xxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx
xxxxxxxx.xxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx
xxxxxxxx.xxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx
xxxxxxxx.xxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx
xxxxxxxx.xxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx
xxxxxxxx.xxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx
xxxxxxxx.xxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx
xxxxxxxx.xxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx
xxxxxxxx.xxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx
xxxxxxxx.xxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx
xxxxxxxx.xxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx
xxxxxxxx.xxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx
xxxxxxxx.xxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx
xxxxxxxx.xxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx
xxxxxxxx.xxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx
xxxxxxxx.xxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx
xxxxxxxx.xxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx
xxxxxxxx.xxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx
```

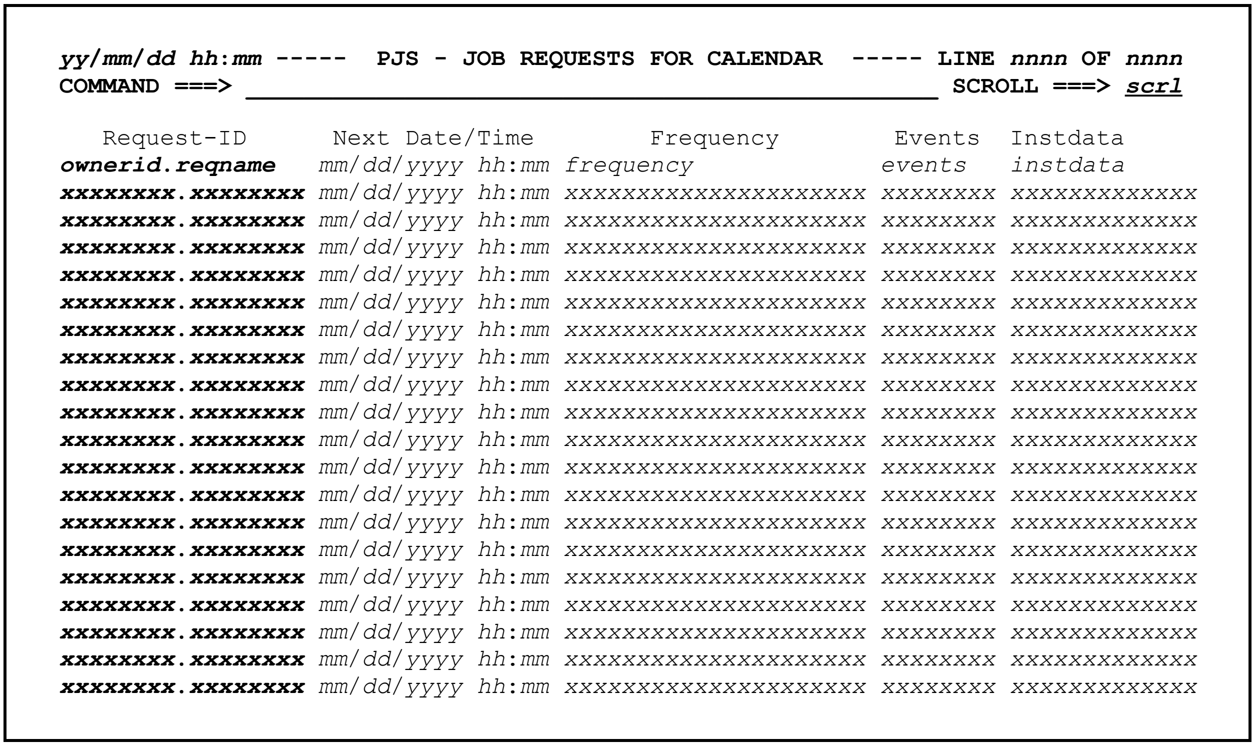
Figure 56: Display Job Requests for Calendar Panel (Part 1 of 3)



```
yy/mm/dd hh:mm -----  PJS - JOB REQUESTS FOR CALENDAR   ----- LINE nnnn OF nnnn
COMMAND ===> ________________________________________________ SCROLL ===> scrl

   Request-ID                JCL Data Set
ownerid.reqname   data.set.name(member)
xxxxxxxx.xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx(xxxxxxxx)
xxxxxxxx.xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx(xxxxxxxx)
xxxxxxxx.xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx(xxxxxxxx)
xxxxxxxx.xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx(xxxxxxxx)
xxxxxxxx.xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx(xxxxxxxx)
xxxxxxxx.xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx(xxxxxxxx)
xxxxxxxx.xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx(xxxxxxxx)
xxxxxxxx.xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx(xxxxxxxx)
xxxxxxxx.xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx(xxxxxxxx)
xxxxxxxx.xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx(xxxxxxxx)
xxxxxxxx.xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx(xxxxxxxx)
xxxxxxxx.xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx(xxxxxxxx)
xxxxxxxx.xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx(xxxxxxxx)
xxxxxxxx.xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx(xxxxxxxx)
xxxxxxxx.xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx(xxxxxxxx)
xxxxxxxx.xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx(xxxxxxxx)
xxxxxxxx.xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx(xxxxxxxx)
xxxxxxxx.xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx(xxxxxxxx)
xxxxxxxx.xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx(xxxxxxxx)
```

Figure 57: Display Job Requests for Calendar Panel (Part 2 of 3)



```
yy/mm/dd hh:mm -----  PJS - JOB REQUESTS FOR CALENDAR   ----- LINE nnnn OF nnnn
COMMAND ===> ________________________________________________ SCROLL ===> scrl

   Request-ID       Next Date/Time         Frequency        Events  Instdata
ownerid.reqname   mm/dd/yyyy hh:mm frequency             events   instdata
xxxxxxxx.xxxxxxxx mm/dd/yyyy hh:mm xxxxxxxxxxxxxxxxxxxxx xxxxxxxx xxxxxxxxxxxxx
xxxxxxxx.xxxxxxxx mm/dd/yyyy hh:mm xxxxxxxxxxxxxxxxxxxxx xxxxxxxx xxxxxxxxxxxxx
xxxxxxxx.xxxxxxxx mm/dd/yyyy hh:mm xxxxxxxxxxxxxxxxxxxxx xxxxxxxx xxxxxxxxxxxxx
xxxxxxxx.xxxxxxxx mm/dd/yyyy hh:mm xxxxxxxxxxxxxxxxxxxxx xxxxxxxx xxxxxxxxxxxxx
xxxxxxxx.xxxxxxxx mm/dd/yyyy hh:mm xxxxxxxxxxxxxxxxxxxxx xxxxxxxx xxxxxxxxxxxxx
xxxxxxxx.xxxxxxxx mm/dd/yyyy hh:mm xxxxxxxxxxxxxxxxxxxxx xxxxxxxx xxxxxxxxxxxxx
xxxxxxxx.xxxxxxxx mm/dd/yyyy hh:mm xxxxxxxxxxxxxxxxxxxxx xxxxxxxx xxxxxxxxxxxxx
xxxxxxxx.xxxxxxxx mm/dd/yyyy hh:mm xxxxxxxxxxxxxxxxxxxxx xxxxxxxx xxxxxxxxxxxxx
xxxxxxxx.xxxxxxxx mm/dd/yyyy hh:mm xxxxxxxxxxxxxxxxxxxxx xxxxxxxx xxxxxxxxxxxxx
xxxxxxxx.xxxxxxxx mm/dd/yyyy hh:mm xxxxxxxxxxxxxxxxxxxxx xxxxxxxx xxxxxxxxxxxxx
xxxxxxxx.xxxxxxxx mm/dd/yyyy hh:mm xxxxxxxxxxxxxxxxxxxxx xxxxxxxx xxxxxxxxxxxxx
xxxxxxxx.xxxxxxxx mm/dd/yyyy hh:mm xxxxxxxxxxxxxxxxxxxxx xxxxxxxx xxxxxxxxxxxxx
xxxxxxxx.xxxxxxxx mm/dd/yyyy hh:mm xxxxxxxxxxxxxxxxxxxxx xxxxxxxx xxxxxxxxxxxxx
xxxxxxxx.xxxxxxxx mm/dd/yyyy hh:mm xxxxxxxxxxxxxxxxxxxxx xxxxxxxx xxxxxxxxxxxxx
xxxxxxxx.xxxxxxxx mm/dd/yyyy hh:mm xxxxxxxxxxxxxxxxxxxxx xxxxxxxx xxxxxxxxxxxxx
xxxxxxxx.xxxxxxxx mm/dd/yyyy hh:mm xxxxxxxxxxxxxxxxxxxxx xxxxxxxx xxxxxxxxxxxxx
xxxxxxxx.xxxxxxxx mm/dd/yyyy hh:mm xxxxxxxxxxxxxxxxxxxxx xxxxxxxx xxxxxxxxxxxxx
xxxxxxxx.xxxxxxxx mm/dd/yyyy hh:mm xxxxxxxxxxxxxxxxxxxxx xxxxxxxx xxxxxxxxxxxxx
xxxxxxxx.xxxxxxxx mm/dd/yyyy hh:mm xxxxxxxxxxxxxxxxxxxxx xxxxxxxx xxxxxxxxxxxxx
```

Figure 58: Display Job Requests for Calendar Panel (Part 3 of 3)

For each job request, the following fields are displayed:

Request-ID
is the **Request-ID** of the job request.

Description
is the **Description** of the job request.

Status
is the **Job Request Status** of the job request. For a list of the possible values, see the section **The Job Request Status** on page 4.

JCL Data Set
is the **User's JCL Data Set** name (and member name, if a PDS is specified).

Next Run D/T
is the **Next Run Date and Time** of the job request.

Frequency
is the **Frequency** of the job request. For a list of the possible Frequency values, see the section **Specifying the Frequency** on page 6.

Events
is a summary of the **Job Request Events** specified for the job request. The following vallues may appear:

NONE	No events are specified.
POSTED	All job request events are currently posted.
xx **OF** *yy*	*yy* events are specified for the job request. Of these, *xx* are currently posted.

Instdata
is installation defined data. This is usually used for security related information, and may not be present. For detailed information on your installation data, see your PJS System Administrator.

The following commands may be entered on the **COMMAND** line:

END	returns to the **Add Calendar** panel, the **Modify Calendar** panel, the **Delete Calendar** panel, the **Display Calendar** panel, or the **List Calendars for Owner** panel.
UP *lines*	scrolls the display up by the specified number of lines.
DOWN *lines*	scrolls the display down by the specified number of lines.
LEFT *cols*	scrolls the display left by the specified number of columns.
RIGHT *cols*	scrolls the display right by the specified number of columns.

Event Panels

The Event panels allow you to post, reset, and display PJS Events.

Event Menu Panel

The **Event Menu** panel provides access to the event panels. Inputs are provided for entering the Owner-ID and Event Name of the event to be updated or displayed.

The **Event Menu** panel is accessed from the **PJS Main Menu** panel by selecting option **E**.

```
yy/mm/dd hh:mm -------  PJS - EVENT MENU  -------------------------------------
OPTION ===> ____________________________________________________________________

Select one of the following functions:

  L  LIST     - List all Events for Owner-ID
  P  POST     - Post an Event
  R  RESET    - Reset an Event
  S  DISPLAY  - Display an Event


Event-ID:
  Owner-ID      ===> ownerid_
  Event Name    ===> evntname       (Required for options A, D, M, and S)


Enter Event-ID (if required) and make a selection, then press ENTER.
```

Figure 59: PJS Event Menu Panel

To use this panel, enter the owner-ID and event name of the event to be updated or displayed. Then, enter the letter corresponding to the option desired in the **OPTION** field and press **ENTER**. To return to the PJS Main Menu panel, enter **END** on the option line (or press the **END** PF-key, usually **PF3**).

The following options may be entered on the **OPTION** line:

L selects the **List Events** for Owner panel.

P selects the **Post Event** panel.

R selects the **Reset Event** panel.

S selects the **Display Event** panel.

END returns to the **PJS Main Menu** panel.

The following record key fields may be entered:

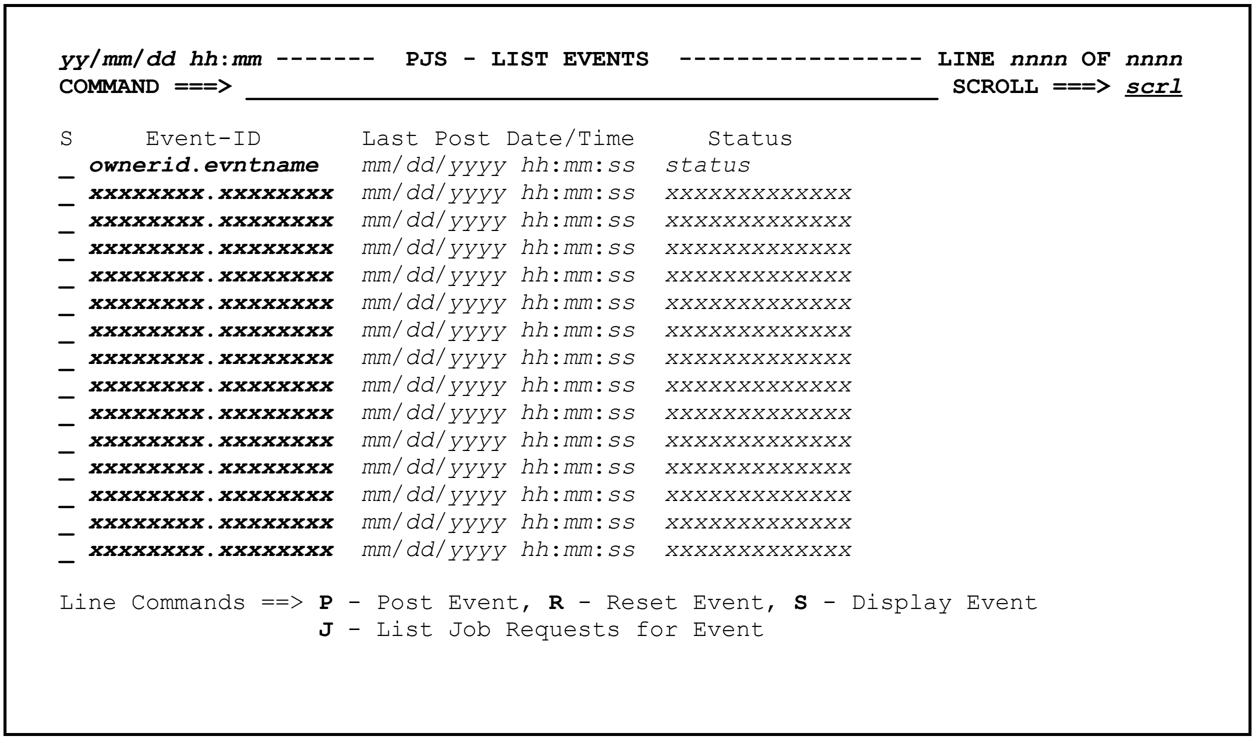
Owner-ID
specifies the **Owner-ID** of the event(s) to be updated or displayed. This field is required.

Event Name
specifies the **Event Name** of the event to be updated or displayed. The format of the event name is described in the section **The Event-ID** on page 24. This field is required when option **P** (Post Event), option **R** (Reset Event), or **S** (Display Event) is selected. It is ignored when option **L** (List Events for Owner) is selected.

List Events for Owner Panel

The **List Events for Owner** panel lists the events owned by the current Owner-ID. The list can be scrolled up and down to display more events. Individual events may then be selected for update or display using line commands.

The **List Events for Owner** panel is accessed from the **Event Menu** panel by selecting option **L**.



```
yy/mm/dd hh:mm -------  PJS - LIST EVENTS  ----------------- LINE nnnn OF nnnn
COMMAND ===> ________________________________________________ SCROLL ===> scrl

S     Event-ID       Last Post Date/Time     Status
_ ownerid.evntname   mm/dd/yyyy hh:mm:ss  status
_ xxxxxxxx.xxxxxxxx  mm/dd/yyyy hh:mm:ss  xxxxxxxxxxxxx
_ xxxxxxxx.xxxxxxxx  mm/dd/yyyy hh:mm:ss  xxxxxxxxxxxxx
_ xxxxxxxx.xxxxxxxx  mm/dd/yyyy hh:mm:ss  xxxxxxxxxxxxx
_ xxxxxxxx.xxxxxxxx  mm/dd/yyyy hh:mm:ss  xxxxxxxxxxxxx
_ xxxxxxxx.xxxxxxxx  mm/dd/yyyy hh:mm:ss  xxxxxxxxxxxxx
_ xxxxxxxx.xxxxxxxx  mm/dd/yyyy hh:mm:ss  xxxxxxxxxxxxx
_ xxxxxxxx.xxxxxxxx  mm/dd/yyyy hh:mm:ss  xxxxxxxxxxxxx
_ xxxxxxxx.xxxxxxxx  mm/dd/yyyy hh:mm:ss  xxxxxxxxxxxxx
_ xxxxxxxx.xxxxxxxx  mm/dd/yyyy hh:mm:ss  xxxxxxxxxxxxx
_ xxxxxxxx.xxxxxxxx  mm/dd/yyyy hh:mm:ss  xxxxxxxxxxxxx
_ xxxxxxxx.xxxxxxxx  mm/dd/yyyy hh:mm:ss  xxxxxxxxxxxxx
_ xxxxxxxx.xxxxxxxx  mm/dd/yyyy hh:mm:ss  xxxxxxxxxxxxx
_ xxxxxxxx.xxxxxxxx  mm/dd/yyyy hh:mm:ss  xxxxxxxxxxxxx
_ xxxxxxxx.xxxxxxxx  mm/dd/yyyy hh:mm:ss  xxxxxxxxxxxxx

Line Commands ==> P - Post Event, R - Reset Event, S - Display Event
                  J - List Job Requests for Event
```

Figure 60: List Events for Owner Panel

For each event, the following fields are displayed:

Event-ID
is the **Event-ID** of the event.

Last Post Date/Time
is the date and time the event was last posted.

Status
is the **Event Status** of the event. For a list of the possible values, see the section **Event Status** on page 11.

The following line commands may be entered in the selection field (under the **'S'** on the left side of the panel) for a calendar:

P selects the **Post Event** panel for the selected event.

R selects the **Reset Event** panel for the selected event.

S selects the **Display Event** panel for the selected event.

J selects the **Display Job Requests for Event** panel for the selected event.

The following commands may be entered on the **COMMAND** line:

END returns to the **Event Menu** panel.

UP *lines* scrolls the display up by the specified number of lines.

DOWN *lines* scrolls the display down by the specified number of lines.

Post Event Panel

The **Post Event** panel is used to post an event. To confirm the post, the user must enter **YES** in the confirmation field. The post is then completed by entering the **END** command. The user may cancel the post by entering the **CANCEL** command.

After posting an event with this panel, the event will be placed in the POST PENDING status. The event will remain in this status briefly until the PJS System Task posts all of the corresponding Job Request Events.

The **Post Event** panel is accessed from the **Event Menu** panel by selecting option **P**, or from the **List Events for Owner** panel by entering a **P** line command on an existing event.

```
yy/mm/dd hh:mm -------  PJS - POST EVENT  -------------------------------------
COMMAND ===> ____________________________________________________________________

Event-ID         ===> ownerid.evntname

Event Status     ===> status

Date/Time Posted ===> mm/dd/yyyy hh:mm:ss


Enter 'YES' and press enter to confirm post ===> ___









Commands ==> REQ - Display Job Requests for Event,
             END - Complete Post, CANCEL - Cancel Post
```

Figure 61: Post Event Panel

The following fields are displayed, but may not be updated:

Event-ID
is the **Event-ID** of the event.

Status
is the **Event Status** of the event. For a list of the possible values, see the section **Event Status** on page 11.

Last Post Date/Time
is the date and time the event was last posted.

The following commands may be entered on the **COMMAND** line:

REQ	displays the **Display Job Requests For Event** panel.
END	completes the post and returns to the **Event Menu** panel, or the **List Events for Owner** panel.
CANCEL	cancels the post and returns to the **Event Menu** panel, or the **List Events for Owner** panel, without updating PJS.

Reset Event Panel

The **Reset Event** panel is used to reset an event. To confirm the reset, the user must enter **YES** in the confirmation field. The reset is then completed by entering the **END** command. The user may cancel the reset by entering the **CANCEL** command.

After reseting an event with this panel, the event will be placed in the RESET PENDING status. The event will remain in this status briefly until the PJS System Task resets all of the corresponding Job Request Events.

The **Reset Event** panel is accessed from the **Event Menu** panel by selecting option **R**, or from the **List Events for Owner** panel by entering an **R** line command on an existing event.

```
yy/mm/dd hh:mm -------  PJS - RESET EVENT  ------------------------------------
COMMAND ===> ____________________________________________________________________

Event-ID         ===> ownerid.evntname

Event Status     ===> status

Date/Time Posted ===> mm/dd/yyyy hh:mm:ss


Enter 'YES' and press enter to confirm reset ===> ___









Commands ==> REQ - Display Job Requests for Event,
             END - Complete Reset, CANCEL - Cancel Reset
```

Figure 62: Reset Event Panel

The following fields are displayed, but may not be updated:

Event-ID
is the **Event-ID** of the event.

Status
is the **Event Status** of the event. For a list of the possible values, see the section **Event Status** on page 11.

Last Post Date/Time
is the date and time the event was last posted.

The following commands may be entered on the **COMMAND** line:

REQ	displays the **Display Job Requests For Event** panel.
END	completes the reset and returns to the **Event Menu** panel, or the **List Events for Owner** panel.
CANCEL	cancels the reset and returns to the **Event Menu** panel, or the **List Events for Owner** panel, without updating PJS.

Display Event Panel

The **Display Event** panel is used to display an event.

The **Display Event** panel is accessed from the **Event Menu** panel by selecting option **S**, or from the **List Events for Owner** panel by entering an **S** line command on an existing event.

```
yy/mm/dd hh:mm -------  PJS - DISPLAY EVENT  ----------------------------------
COMMAND ===> ____________________________________________________________________

Event-ID         ===> ownerid.evntname

Event Status     ===> status

Date/Time Posted ===> mm/dd/yyyy hh:mm:ss









Commands ==> REQ - Display Job Requests for Event
```

Figure 63: Display Event Panel

The following fields are displayed:

Event-ID
is the **Event-ID** of the event.

Status
is the **Event Status** of the event. For a list of the possible values, see the section **Event Status** on page 11.

Last Post Date/Time
is the date and time the event was last posted.

The following commands may be entered on the **COMMAND** line:

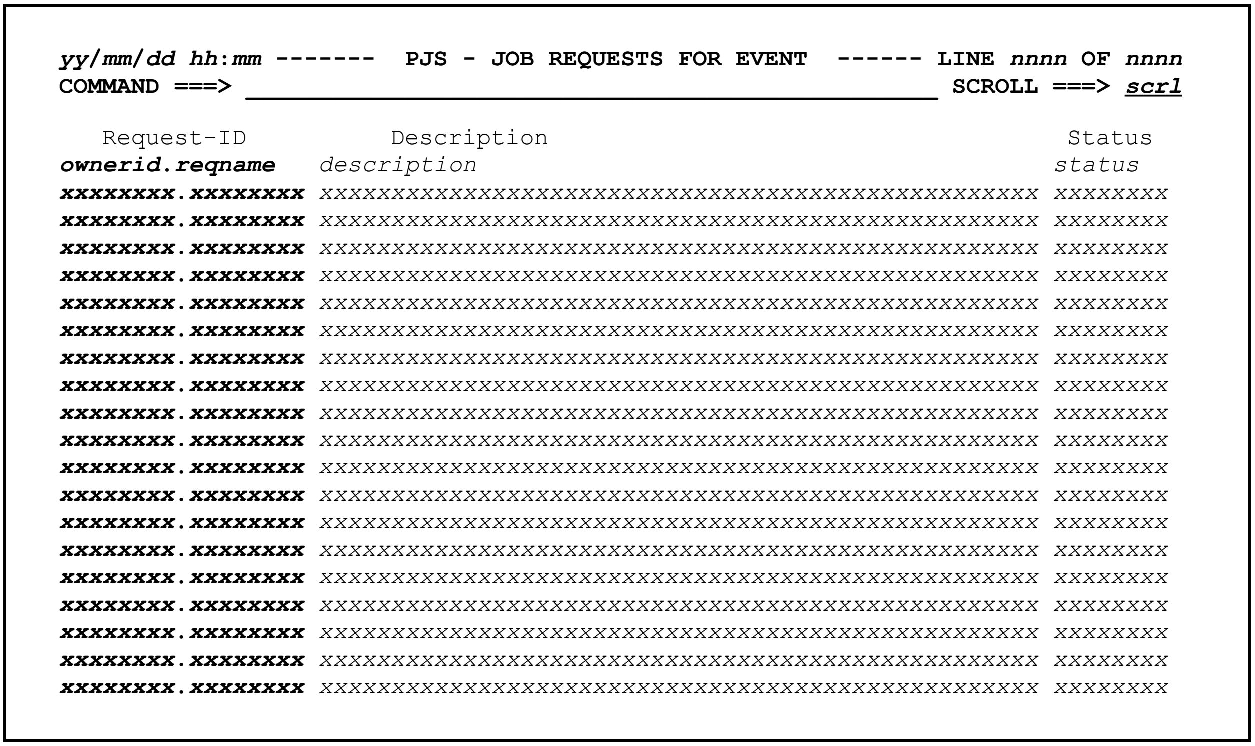
REQ displays the **Display Job Requests For Event** panel.

END	returns to the **Event Menu** panel, or the **List Events for Owner** panel.

List Job Requests for Event Panel

The **List Job Requests for Event** panel lists the job requests that specify an event. The list can be scrolled up and down to display more job requests, and left and right to display additional information for each job request.

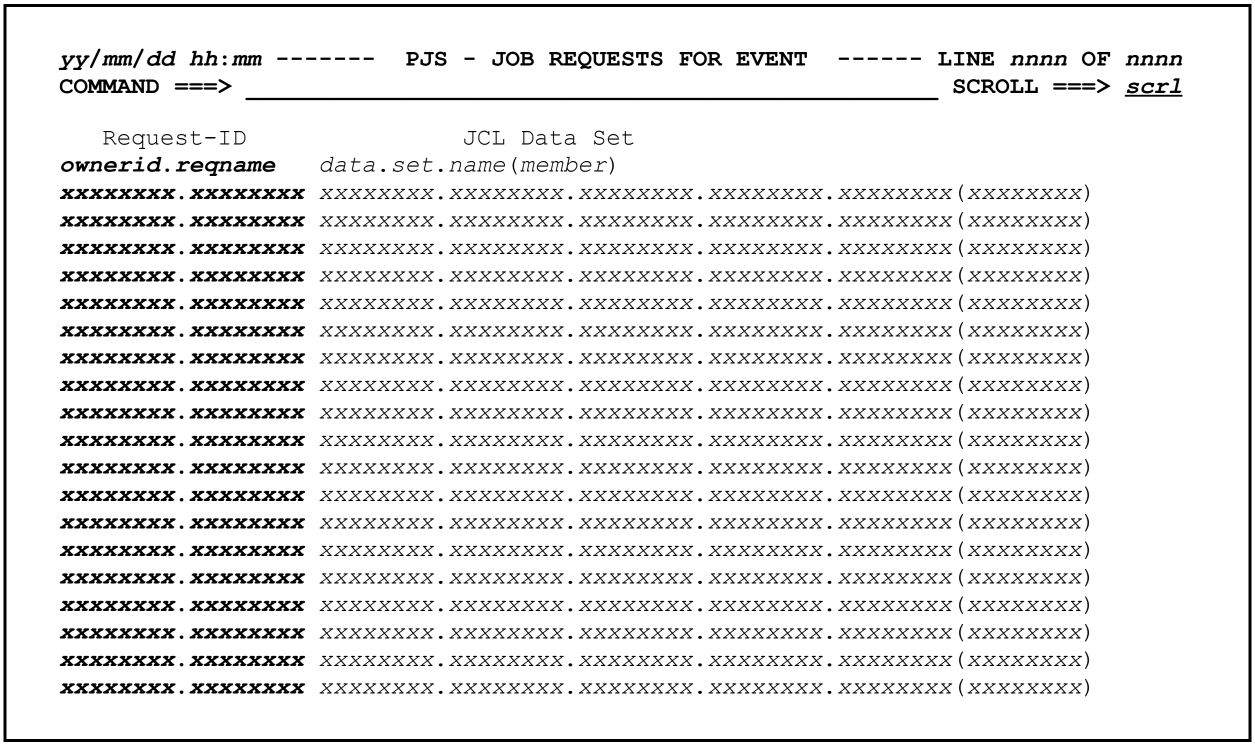
The **List Job Requests for Event** panel is accessed from the **Post Event** panel, the **Reset Event** panel, or the **Display Event** panel, by entering the **REQ** command on the command line, or from the **List Events for Owner** panel by entering a **J** line command on an event.



```
yy/mm/dd hh:mm -------  PJS - JOB REQUESTS FOR EVENT  ------ LINE nnnn OF nnnn
COMMAND ===> ________________________________________________ SCROLL ===> scrl

   Request-ID          Description                                     Status
ownerid.reqname   description                                         status
xxxxxxxx.xxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx
xxxxxxxx.xxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx
xxxxxxxx.xxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx
xxxxxxxx.xxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx
xxxxxxxx.xxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx
xxxxxxxx.xxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx
xxxxxxxx.xxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx
xxxxxxxx.xxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx
xxxxxxxx.xxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx
xxxxxxxx.xxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx
xxxxxxxx.xxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx
xxxxxxxx.xxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx
xxxxxxxx.xxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx
xxxxxxxx.xxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx
xxxxxxxx.xxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx
xxxxxxxx.xxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx
xxxxxxxx.xxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx
xxxxxxxx.xxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx
xxxxxxxx.xxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx
```

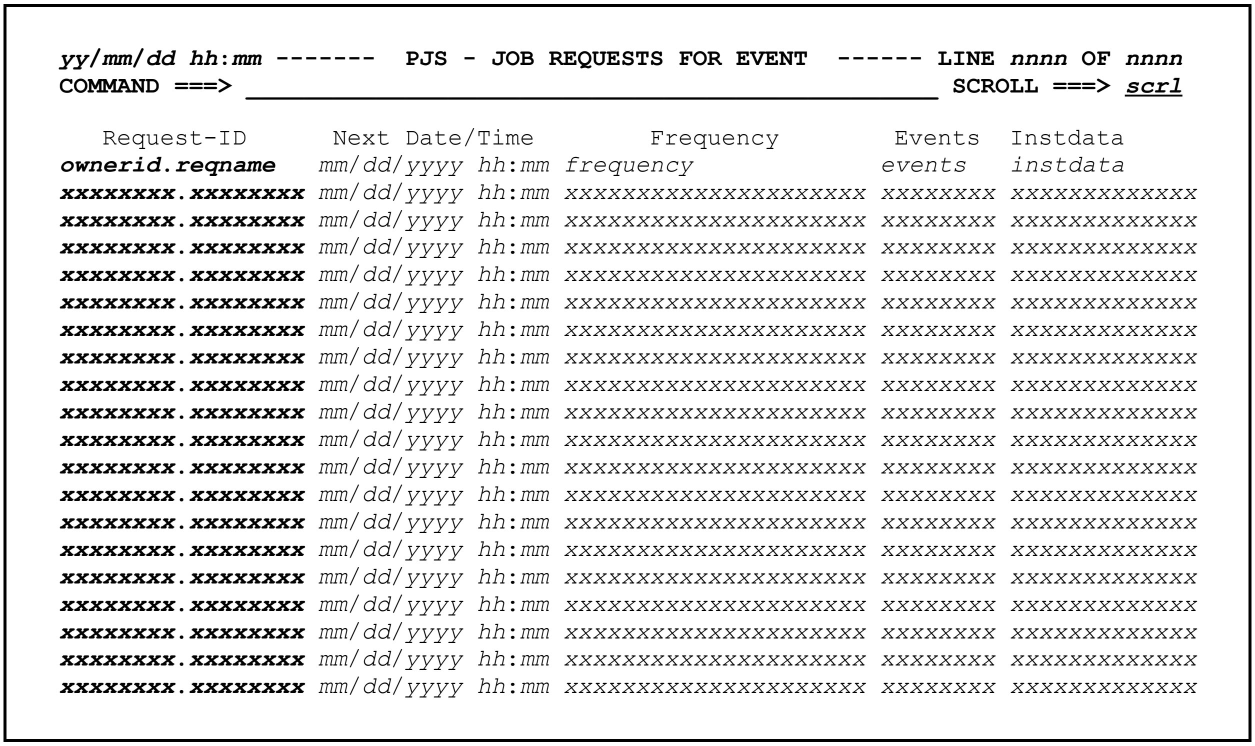
Figure 64: List Job Requests for Event Panel (Part 1 of 3)



```
yy/mm/dd hh:mm -------  PJS - JOB REQUESTS FOR EVENT  ------ LINE nnnn OF nnnn
COMMAND ===> ________________________________________________ SCROLL ===> scrl

   Request-ID               JCL Data Set
ownerid.reqname   data.set.name(member)
xxxxxxxx.xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx(xxxxxxxx)
xxxxxxxx.xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx(xxxxxxxx)
xxxxxxxx.xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx(xxxxxxxx)
xxxxxxxx.xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx(xxxxxxxx)
xxxxxxxx.xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx(xxxxxxxx)
xxxxxxxx.xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx(xxxxxxxx)
xxxxxxxx.xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx(xxxxxxxx)
xxxxxxxx.xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx(xxxxxxxx)
xxxxxxxx.xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx(xxxxxxxx)
xxxxxxxx.xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx(xxxxxxxx)
xxxxxxxx.xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx(xxxxxxxx)
xxxxxxxx.xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx(xxxxxxxx)
xxxxxxxx.xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx(xxxxxxxx)
xxxxxxxx.xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx(xxxxxxxx)
xxxxxxxx.xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx(xxxxxxxx)
xxxxxxxx.xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx(xxxxxxxx)
xxxxxxxx.xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx(xxxxxxxx)
xxxxxxxx.xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx(xxxxxxxx)
xxxxxxxx.xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx(xxxxxxxx)
```

Figure 65: List Job Requests for Event Panel (Part 2 of 3)



```
yy/mm/dd hh:mm -------  PJS - JOB REQUESTS FOR EVENT  ------ LINE nnnn OF nnnn
COMMAND ===> ________________________________________________ SCROLL ===> scrl

   Request-ID      Next Date/Time        Frequency        Events  Instdata
ownerid.reqname   mm/dd/yyyy hh:mm frequency             events   instdata
xxxxxxxx.xxxxxxxx mm/dd/yyyy hh:mm xxxxxxxxxxxxxxxxxxxxx xxxxxxxx xxxxxxxxxxxxx
xxxxxxxx.xxxxxxxx mm/dd/yyyy hh:mm xxxxxxxxxxxxxxxxxxxxx xxxxxxxx xxxxxxxxxxxxx
xxxxxxxx.xxxxxxxx mm/dd/yyyy hh:mm xxxxxxxxxxxxxxxxxxxxx xxxxxxxx xxxxxxxxxxxxx
xxxxxxxx.xxxxxxxx mm/dd/yyyy hh:mm xxxxxxxxxxxxxxxxxxxxx xxxxxxxx xxxxxxxxxxxxx
xxxxxxxx.xxxxxxxx mm/dd/yyyy hh:mm xxxxxxxxxxxxxxxxxxxxx xxxxxxxx xxxxxxxxxxxxx
xxxxxxxx.xxxxxxxx mm/dd/yyyy hh:mm xxxxxxxxxxxxxxxxxxxxx xxxxxxxx xxxxxxxxxxxxx
xxxxxxxx.xxxxxxxx mm/dd/yyyy hh:mm xxxxxxxxxxxxxxxxxxxxx xxxxxxxx xxxxxxxxxxxxx
xxxxxxxx.xxxxxxxx mm/dd/yyyy hh:mm xxxxxxxxxxxxxxxxxxxxx xxxxxxxx xxxxxxxxxxxxx
xxxxxxxx.xxxxxxxx mm/dd/yyyy hh:mm xxxxxxxxxxxxxxxxxxxxx xxxxxxxx xxxxxxxxxxxxx
xxxxxxxx.xxxxxxxx mm/dd/yyyy hh:mm xxxxxxxxxxxxxxxxxxxxx xxxxxxxx xxxxxxxxxxxxx
xxxxxxxx.xxxxxxxx mm/dd/yyyy hh:mm xxxxxxxxxxxxxxxxxxxxx xxxxxxxx xxxxxxxxxxxxx
xxxxxxxx.xxxxxxxx mm/dd/yyyy hh:mm xxxxxxxxxxxxxxxxxxxxx xxxxxxxx xxxxxxxxxxxxx
xxxxxxxx.xxxxxxxx mm/dd/yyyy hh:mm xxxxxxxxxxxxxxxxxxxxx xxxxxxxx xxxxxxxxxxxxx
xxxxxxxx.xxxxxxxx mm/dd/yyyy hh:mm xxxxxxxxxxxxxxxxxxxxx xxxxxxxx xxxxxxxxxxxxx
xxxxxxxx.xxxxxxxx mm/dd/yyyy hh:mm xxxxxxxxxxxxxxxxxxxxx xxxxxxxx xxxxxxxxxxxxx
xxxxxxxx.xxxxxxxx mm/dd/yyyy hh:mm xxxxxxxxxxxxxxxxxxxxx xxxxxxxx xxxxxxxxxxxxx
xxxxxxxx.xxxxxxxx mm/dd/yyyy hh:mm xxxxxxxxxxxxxxxxxxxxx xxxxxxxx xxxxxxxxxxxxx
xxxxxxxx.xxxxxxxx mm/dd/yyyy hh:mm xxxxxxxxxxxxxxxxxxxxx xxxxxxxx xxxxxxxxxxxxx
xxxxxxxx.xxxxxxxx mm/dd/yyyy hh:mm xxxxxxxxxxxxxxxxxxxxx xxxxxxxx xxxxxxxxxxxxx
```

Figure 66: List Job Requests for Event Panel (Part 3 of 3)

For each job request, the following fields are displayed:

Request-ID
is the **Request-ID** of the job request.

Description
is the **Description** of the job request.

Status
is the **Job Request Status** of the job request. For a list of the possible values, see the section **The Job Request Status** on page 4.

JCL Data Set
is the **User's JCL Data Set** name (and member name, if a PDS is specified).

Next Run D/T
is the **Next Run Date and Time** of the job request.

Frequency
is the **Frequency** of the job request. For a list of the possible Frequency values, see the section **Specifying the Frequency** on page 6.

Events
is a summary of the **Job Request Events** specified for the job request. The following vallues may appear:

NONE	No events are specified.
POSTED	All job request events are currently posted.
xx **OF** *yy*	*yy* events are specified for the job request. Of these, *xx* are currently posted.

Instdata
is installation defined data. This is usually used for security related information, and may not be present. For detailed information on your installation data, see your PJS System Administrator.

The following commands may be entered on the **COMMAND** line:

END	returns to the **Post Event** panel, the **Reset Event** panel, the **Display Event** panel, or the **List Events for Owner** panel.
UP *lines*	scrolls the display up by the specified number of lines.
DOWN *lines*	scrolls the display down by the specified number of lines.
LEFT *cols*	scrolls the display left by the specified number of columns.
RIGHT *cols*	scrolls the display right by the specified number of columns.

Global Variable Panels

The Global Variable panels allow you to add, modify, delete, and display PJS Global Variables.

Global Variable Menu Panel

The **Global Variable Menu** panel provides access to the global variable panels. Inputs are provided for entering the Owner-ID and Variable Name of the global variable to be updated or displayed.

The **Global Variable Menu** panel is accessed from the **PJS Main Menu** panel by selecting option **V**.

```
yy/mm/dd hh:mm -------  PJS - VARIABLE MENU  ----------------------------------
OPTION ===> ____________________________________________________________________

Select one of the following functions:

  L  LIST     - List all Variables for Owner-ID
  A  ADD      - Add a new Variables
  D  DELETE   - Delete a Variable
  M  MODIFY   - Modify a Variable
  S  DISPLAY  - Display a Variable


Variable-ID:
  Owner-ID      ===> ownerid_
  Variable Name ===> varname_       (Required for options A, D, M, and S)


Enter Variable-ID (if required) and make a selection, then press ENTER.
```

Figure 67: PJS Global Variable Menu Panel

To use this panel, enter the owner-ID and variable name of the global variable to be updated or displayed. Then, enter the letter corresponding to the option desired in the **OPTION** field and press **ENTER**. To return to the PJS Main Menu panel, enter **END** on the option line (or press the **END** PF-key, usually **PF3**).

The following options may be entered on the **OPTION** line:

A selects the **Add Global Variable** panel.

D selects the **Delete Global Variable** panel.

L selects the **List Global Variables** for Owner panel.

M selects the **Modify Global Variable** panel.

S selects the **Display Global Variable** panel.

END returns to the **PJS Main Menu** panel.

The following record key fields may be entered:

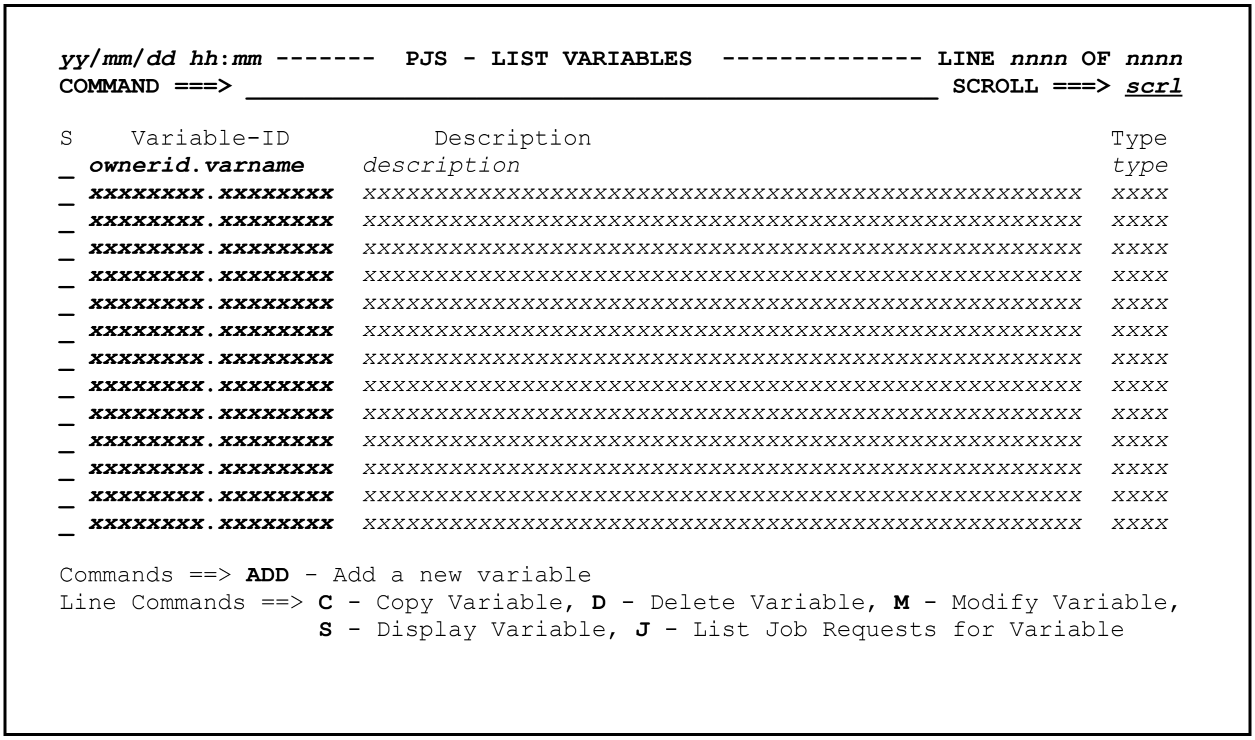
Owner-ID
specifies the **Owner-ID** of the global variable(s) to be updated or displayed. This field is required.

Variable Name
specifies the **Variable Name** of the global variable to be updated or displayed. The format of the variable name is described in the section **The Global Variable-ID** on page 25. This field is required when option **A** (Add Global Variable), option **D** (Delete Global Variable), **M** (Modify Global Variable), or **S** (Display Global Variable) is selected. It is ignored when option **L** (List Global Variables for Owner) is selected.

List Global Variables Panel

The **List Global Variables for Owner** panel lists the global variables owned by the current Owner-ID. The list can be scrolled up and down to display more global variables, and left and right to display additional information for each global variable. Individual global variables may then be selected for update or display using line commands.

The **List Global Variables for Owner** panel is accessed from the **Global Variable Menu** panel by selecting option **L**.

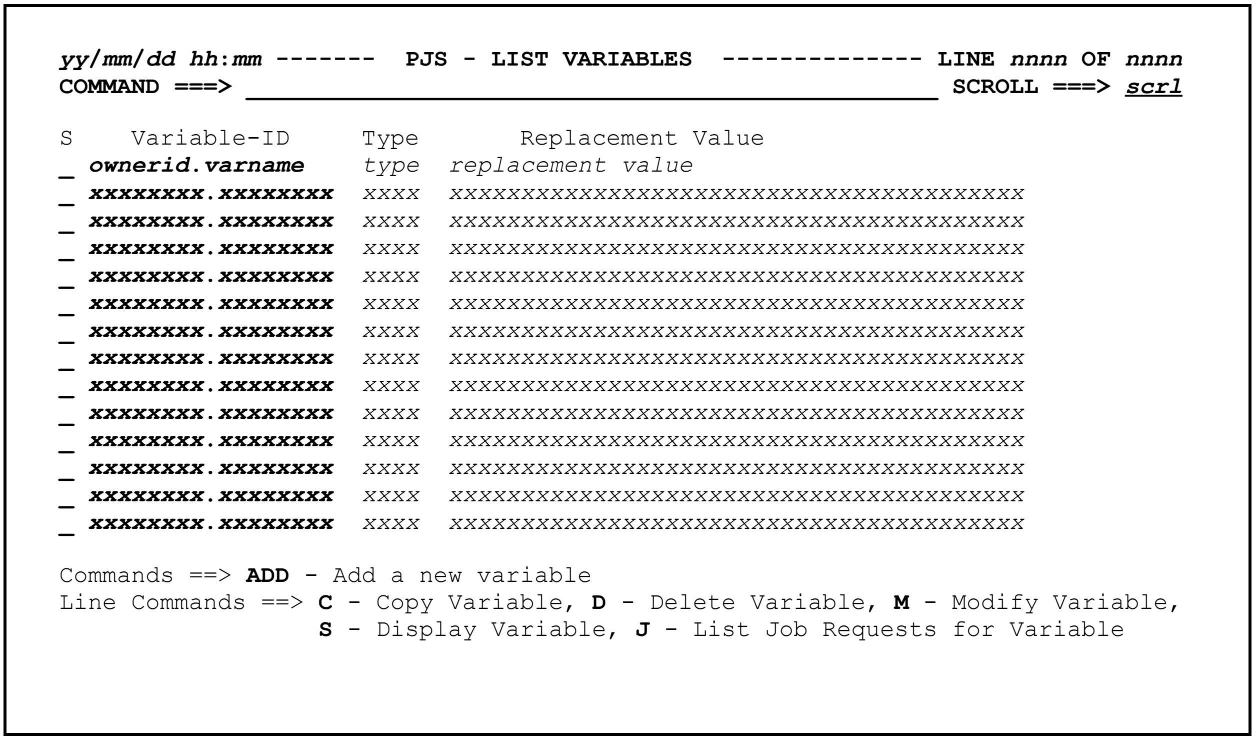


```
yy/mm/dd hh:mm -------  PJS - LIST VARIABLES  -------------- LINE nnnn OF nnnn
COMMAND ===> ________________________________________________ SCROLL ===> scrl

S    Variable-ID          Description                                     Type
_ ownerid.varname    description                                          type
_ xxxxxxxx.xxxxxxxx  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx  xxxx
_ xxxxxxxx.xxxxxxxx  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx  xxxx
_ xxxxxxxx.xxxxxxxx  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx  xxxx
_ xxxxxxxx.xxxxxxxx  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx  xxxx
_ xxxxxxxx.xxxxxxxx  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx  xxxx
_ xxxxxxxx.xxxxxxxx  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx  xxxx
_ xxxxxxxx.xxxxxxxx  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx  xxxx
_ xxxxxxxx.xxxxxxxx  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx  xxxx
_ xxxxxxxx.xxxxxxxx  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx  xxxx
_ xxxxxxxx.xxxxxxxx  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx  xxxx
_ xxxxxxxx.xxxxxxxx  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx  xxxx
_ xxxxxxxx.xxxxxxxx  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx  xxxx
_ xxxxxxxx.xxxxxxxx  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx  xxxx

Commands ==> ADD - Add a new variable
Line Commands ==> C - Copy Variable, D - Delete Variable, M - Modify Variable,
                  S - Display Variable, J - List Job Requests for Variable
```

Figure 68: List Global Variables Panel (Part 1 of 2)



```
yy/mm/dd hh:mm -------  PJS - LIST VARIABLES  -------------- LINE nnnn OF nnnn
COMMAND ===> _______________________________________________ SCROLL ===> scrl

S    Variable-ID     Type       Replacement Value
_ ownerid.varname    type  replacement value
_ xxxxxxxx.xxxxxxxx  xxxx  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
_ xxxxxxxx.xxxxxxxx  xxxx  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
_ xxxxxxxx.xxxxxxxx  xxxx  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
_ xxxxxxxx.xxxxxxxx  xxxx  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
_ xxxxxxxx.xxxxxxxx  xxxx  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
_ xxxxxxxx.xxxxxxxx  xxxx  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
_ xxxxxxxx.xxxxxxxx  xxxx  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
_ xxxxxxxx.xxxxxxxx  xxxx  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
_ xxxxxxxx.xxxxxxxx  xxxx  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
_ xxxxxxxx.xxxxxxxx  xxxx  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
_ xxxxxxxx.xxxxxxxx  xxxx  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
_ xxxxxxxx.xxxxxxxx  xxxx  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
_ xxxxxxxx.xxxxxxxx  xxxx  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

Commands ==> ADD - Add a new variable
Line Commands ==> C - Copy Variable, D - Delete Variable, M - Modify Variable,
                  S - Display Variable, J - List Job Requests for Variable
```

Figure 69: List Global Variables Panel (Part 2 of 2)

For each global variable, the following fields are displayed:

Variable-ID
is the **Variable-ID** of the global variable.

Description
is the **Description** of the global variable.

Type
is the **type** of the **Replacement Value** for the global variable. This may be one of the following:

LIT	The replacement value is a **Literal Value**.
DYN	The replacement value is a **Dynamic Value**.
VAR	The replacement value is another **Global Variable**.

Replacement Value
is the **Replacement Value** of the global variable. The format of the replacement value depends on the **Replacement Value Type**.

When the Replacement Value Type is **LIT** (Literal) this is a literal string.

When the Replacement Value Type is **DYN** (Dynamic Value), this is the name of the dynamic value to be used. Dynamic values may be defined by PJS, or determined by an installation provided routine. The dynamic values provided by PJS are described in the section **Using Dynamic Values** on page 14. For a list of any installation provided routines, see your PJS System Administrator.

When the Replacement Value Type is **VAR** (Global Variable), this is the Variable-ID of another global variable whose value is to be used as the replacement value.

The following line commands may be entered in the selection field (under the **'S'** on the left side of the panel) for a global variable:

C selects the **Add Global Variable** panel to add a new global variable using the selected global variable as a model.

D selects the **Delete Global Variable** panel for the selected global variable.

M selects the **Modify Global Variable** panel for the selected global variable.

S selects the **Display Global Variable** panel for the selected global variable.

J selects the **Display Job Requests for Global Variable** panel for the selected global variable.

The following commands may be entered on the **COMMAND** line:

ADD selects the **Add Global Variable** panel for a new calendar.

END returns to the **Global Variable Menu** panel.

UP *lines* scrolls the display up by the specified number of lines.

DOWN *lines* scrolls the display down by the specified number of lines.

LEFT *cols* scrolls the display left by the specified number of columns.

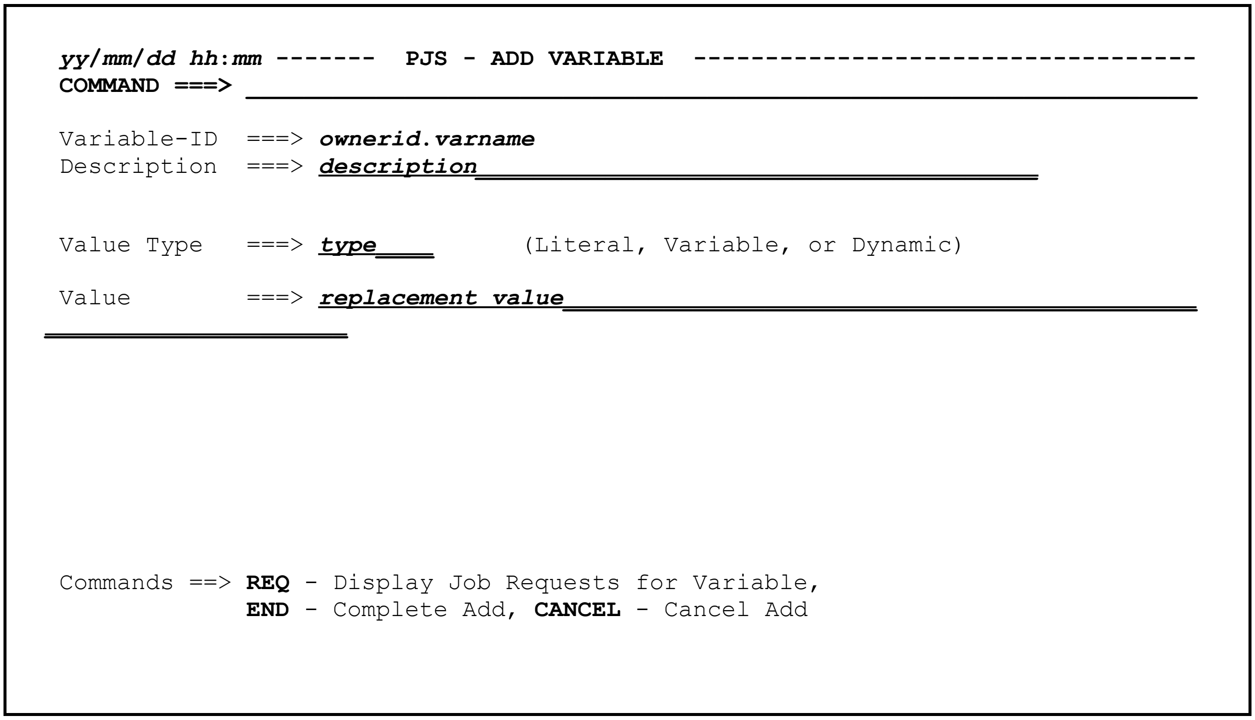
RIGHT *cols* scrolls the display right by the specified number of columns.

Add Global Variable Panel

The **Add Global Variable** panel is used to add a new global variable. When this panel is first displayed, a skeleton global variable containing only default information is created in the ISPF user's storage. The user then enters the desired parameters. The add is not completed until the **END** command is entered, indicating that all parameters have been entered. At any time prior to entering the END command, the user may cancel the add by entering the **CANCEL** command.

The **Add Global Variable** panel is accessed from the **Global Variable Menu** panel by selecting option **A**, or from the **List Global Variables for Owner** panel by either entering the **ADD** command on the command line, or by entering a **C** line command on an existing global variable.

If the **Add Global Variable** dialog is called from the **List Global Variables for Owner** panel, the **Add Global Variable-ID** panel will be displayed first, to prompt the user for the Variable-ID to be added.



```
yy/mm/dd hh:mm -------  PJS - ADD VARIABLE  ----------------------------------
COMMAND ===> _________________________________________________________________

Variable-ID  ===> ownerid.varname
Description  ===> description_______________________________________


Value Type   ===> type____      (Literal, Variable, or Dynamic)

Value        ===> replacement value___________________________________________
_____________________









Commands ==> REQ - Display Job Requests for Variable,
             END - Complete Add, CANCEL - Cancel Add
```

Figure 70: Add Global Variable Panel

The following fields are displayed, but may not be updated:

Variable-ID
is the **Variable-ID** of the global variable being added. This parameter cannot be changed.

The following data fields may be updated

Description

specifies the **Description** of the global variable. This is for documentation purposes only and does not affect PJS processing in any way.

Value Type

specifies the **type** of the **Replacement Value** for the global variable. This may be one of the following:

LIT The replacement value is a **Literal Value**.

DYN The replacement value is a **Dynamic Value**.

VAR The replacement value is another **Global Variable**.

Value

specifies the **Replacement Value** of the global variable. The format of the replacement value depends on the **Replacement Value Type**.

When the Replacement Value Type is **LIT** (Literal) this is a literal string. If quotes are not entered they will be automatically added.

When the Replacement Value Type is **DYN** (Dynamic Value), this is the 1 to 16 character name of the dynamic value to be used. Dynamic values may be defined by PJS, or determined by an installation provided routine. The dynamic values provided by PJS are described in the section **Using Dynamic Values** on page 14. For a list of any installation provided routines, see your PJS System Administrator.

When the Replacement Value Type is **VAR** (Global Variable), this is the variable-ID of another global variable whose value is to be used as the replacement value. The format of a global variable-ID is described in the section **The Global Variable-ID** on page 25.

The following commands may be entered on the **COMMAND** line:

REQ displays the **Display Job Requests For Global Variable** panel.

END completes the add and returns to the **Global Variable Menu** panel, or the **List Global Variables for Owner** panel.

CANCEL cancels the add and returns to the **Global Variable Menu** panel, or the **List Global Variables for Owner** panel, without updating PJS.

Add Variable-ID Panel

The **Add Variable-ID** panel is used to specify the Variable-ID when the **Add Global Variable** dialog is entered from the **List Global Variables for Owner** panel. This panel is displayed before the **Add Global Variable** panel. After this panel is completed, the **Add Global Variable** panel will be displayed.

```
yy/mm/dd hh:mm -------  PJS - ADD VARIABLE ID  --------------------------------
COMMAND ===> ____________________________________________________________________


New Variable-ID:
  Owner-ID      ===> ownerid_
  Variable Name ===> varname_


Enter New Variable-ID, then press ENTER.









Commands ==> CANCEL - Cancel Add
```

Figure 71: Add Variable ID Panel

The following record key fields may be entered

Owner-ID
specifies the **Owner-ID** of the global variable to be added. This field is required.

Variable Name
specifies the **Variable Name** of the global variable to be added. The format of the variable name is described in the section **The Global Variable-ID** on page 25. This field is required.

The following commands may be entered on the **COMMAND** line:

END returns to the **List Global Variables for Owner** panel.

CANCEL returns to the **List Global Variables for Owner** panel.

Modify Global Variable Panel

The **Modify Global Variable** panel is used to modify an existing global variable. When this panel is first displayed, the global variable is read from the PJS Request Queue and saved in the ISPF user's storage. The user then enters the desired parameters. The modify is not completed until the **END** command is entered, indicating that all parameters have been entered. At any time prior to entering the END command, the user may cancel the modify by entering the **CANCEL** command.

The **Modify Global Variable** panel is accessed from the **Global Variable Menu** panel by selecting option **M**, or from the **List Global Variables for Owner** panel by entering an **M** line command on an existing global variable.

```
yy/mm/dd hh:mm -------  PJS - MODIFY VARIABLE  --------------------------------
COMMAND ===> ____________________________________________________________________

Variable-ID  ===> ownerid.varname
Description  ===> description_____________________________________

Value Type   ===> type____      (Literal, Variable, or Dynamic)

Value        ===> replacement value_____________________________________________
________________________

Commands ==> REQ - Display Job Requests for Variable,
             END - Complete Modify, CANCEL - Cancel Modify
```

Figure 72: Modify Global Variable Panel

The following fields are displayed, but may not be updated:

Variable-ID
is the **Variable-ID** of the global variable being modified. This parameter cannot be changed.

The following data fields may be updated

Description

specifies the **Description** of the global variable. This is for documentation purposes only and does not affect PJS processing in any way.

Value Type

specifies the **type** of the **Replacement Value** for the global variable. This may be one of the following:

LIT The replacement value is a **Literal Value**.

DYN The replacement value is a **Dynamic Value**.

VAR The replacement value is another **Global Variable**.

Value

specifies the **Replacement Value** of the global variable. The format of the replacement value depends on the **Replacement Value Type**.

When the Replacement Value Type is **LIT** (Literal) this is a literal string. If quotes are not entered they will be automatically added.

When the Replacement Value Type is **DYN** (Dynamic Value), this is the 1 to 16 character name of the dynamic value to be used. Dynamic values may be defined by PJS, or determined by an installation provided routine. The dynamic values provided by PJS are described in the section **Using Dynamic Values** on page 14. For a list of any installation provided routines, see your PJS System Administrator.

When the Replacement Value Type is **VAR** (Global Variable), this is the Variable-ID of another global variable whose value is to be used as the replacement value. The format of a global variable-ID is described in the section **The Global Variable-ID** on page 25.

The following commands may be entered on the **COMMAND** line:

REQ displays the **Display Job Requests For Global Variable** panel.

END completes the modify and returns to the **Global Variable Menu** panel, or the **List Global Variables for Owner** panel.

CANCEL cancels the modify and returns to the **Global Variable Menu** panel, or the **List Global Variables for Owner** panel, without updating PJS.

Delete Global Variable Panel

The **Delete Global Variable** panel is used to delete a global variable. To confirm the delete, the user must enter **YES** in the confirmation field. The delete is then completed by entering the **END** command. The user may cancel the delete by entering the **CANCEL** command.

The user should beware that it is possible to delete a global variable that is currently being used by a job request. If this is done, there will be no indication until the next time the PJS System Task attempts to submit the job for the job request. The job submit will fail with message PJS1601E, and the job request will be placed in the ERROR status. To determine which job requests use a global variable, use the **List Job Requests for Global Variable** ISPF dialog.

The **Delete Global Variable** panel is accessed from the **Global Variable Menu** panel by selecting option **D**, or from the **List Global Variables for Owner** panel by entering a **D** line command on an existing global variable.

```
yy/mm/dd hh:mm -------  PJS - DELETE VARIABLE  --------------------------------
COMMAND ===> ____________________________________________________________________

Variable-ID  ===> ownerid.varname
Description  ===> description


Value Type   ===> type

Value        ===> replacement value



Enter 'YES' and press enter to confirm delete ===> ___




Commands ==> REQ - Display Job Requests for Variable,
             END - Complete Delete, CANCEL - Cancel Delete
```

Figure 73: Delete Global Variable Panel

The following fields are displayed, but may not be updated:

Variable-ID
is the **Variable-ID** of the global variable being modified. This parameter cannot be updated.

Description
is the **Description** of the global variable.

Value Type
is the **type** of the **Replacement Value** for the global variable. This may be one of the following:

LIT The replacement value is a **Literal Value**.

DYN The replacement value is a **Dynamic Value**.

VAR The replacement value is another **Global Variable**.

Value
is the **Replacement Value** of the global variable. The format of the replacement value depends on the **Replacement Value Type**.

When the Replacement Value Type is **LIT** (Literal) this is a literal string.

When the Replacement Value Type is **DYN** (Dynamic Value), this is the 1 to 16 character name of the dynamic value to be used. Dynamic values may be defined by PJS, or determined by an installation provided routine. The dynamic values provided by PJS are described in the section **Using Dynamic Values** on page 14. For a list of any installation provided routines, see your PJS System Administrator.

When the Replacement Value Type is **VAR** (Global Variable), this is the Variable-ID of another global variable whose value is to be used as the replacement value.

The following commands may be entered on the **COMMAND** line:

REQ displays the **Display Job Requests For Global Variable** panel.

END completes the delete and returns to the **Global Variable Menu** panel, or the **List Global Variables for Owner** panel.

CANCEL cancels the delete and returns to the **Global Variable Menu** panel, or the **List Global Variables for Owner** panel, without updating PJS.

Display Global Variable Panel

The **Display Global Variable** panel is used to delete a global variable.

The **Display Global Variable** panel is accessed from the **Global Variable Menu** panel by selecting option **S**, or from the **List Global Variables for Owner** panel by entering an **S** line command on an existing global variable.

```
yy/mm/dd hh:mm -------  PJS - DISPLAY VARIABLE  -------------------------------
COMMAND ===> ____________________________________________________________________

Variable-ID  ===> ownerid.varname
Description  ===> description


Value Type   ===> type

Value        ===> replacement value









Commands ==> REQ - Display Job Requests for Variable
```

Figure 74: Display Global Variable Panel

The following fields are displayed:

Variable-ID
is the **Variable-ID** of the global variable.

Description
is the **Description** of the global variable.

Value Type
is the **type** of the **Replacement Value** for the global variable. This may be one of the following:

LIT The replacement value is a **Literal Value**.

DYN The replacement value is a **Dynamic Value**.

VAR The replacement value is another **Global Variable**.

Value

is the **Replacement Value** of the global variable. The format of the replacement value depends on the **Replacement Value Type**.

When the Replacement Value Type is **LIT** (Literal) this is a literal string.

When the Replacement Value Type is **DYN** (Dynamic Value), this is the 1 to 16 character name of the dynamic value to be used. Dynamic values may be defined by PJS, or determined by an installation provided routine. The dynamic values provided by PJS are described in the section **Using Dynamic Values** on page 14. For a list of any installation provided routines, see your PJS System Administrator.

When the Replacement Value Type is **VAR** (Global Variable), this is the Variable-ID of another global variable whose value is to be used as the replacement value.

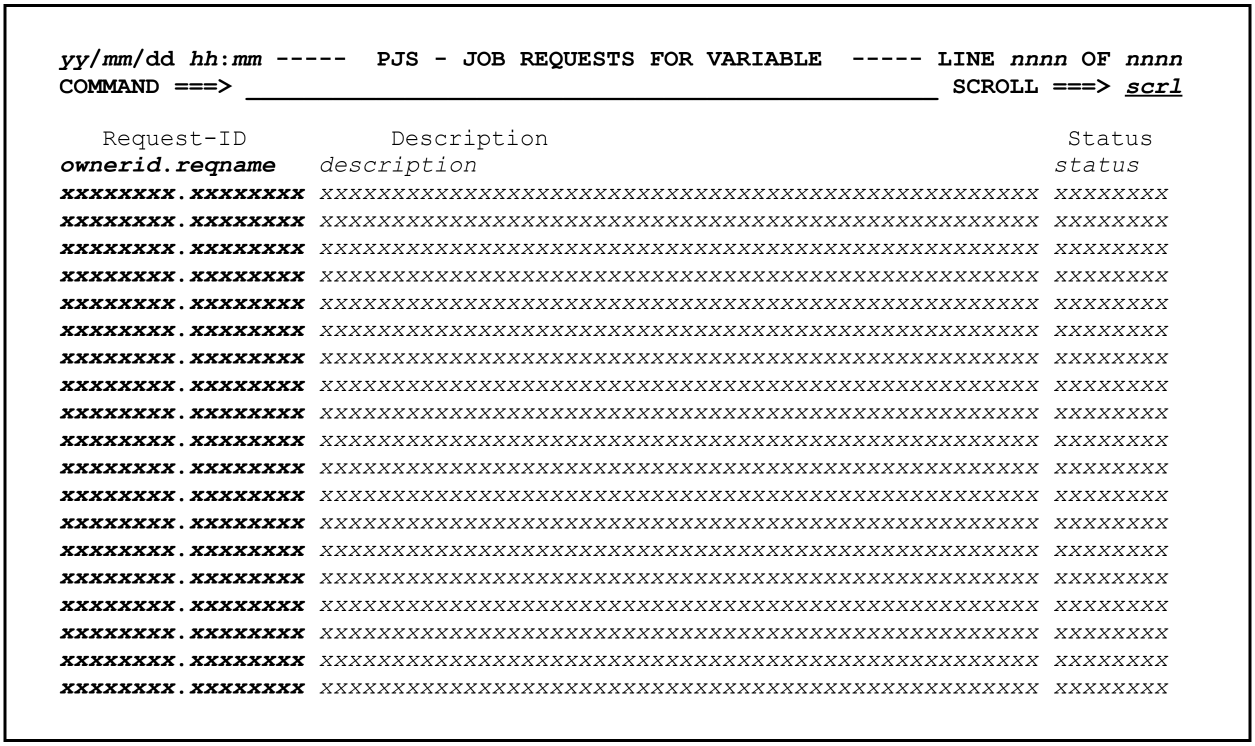
The following commands may be entered on the **COMMAND** line:

REQ	displays the **Display Job Requests For Global Variable** panel.
END	returns to the **Global Variable Menu** panel, or the **List Global Variables for Owner** panel.

List Job Requests for Global Variable Panel

The **List Job Requests for Global Variable** panel lists the job requests that specify a global variable. The list can be scrolled up and down to display more job requests, and left and right to display additional information for each job request.

The **List Job Requests for Global Variable** panel is accessed from the **Add Global Variable** panel, the **Modify Global Variable** panel, the **Delete Global Variable** panel, or the **Display Global Variable** panel, by entering the **REQ** command on the command line, or from the **List Global Variables for Owner** panel by entering a **J** line command on an existing global variable.



```
yy/mm/dd hh:mm -----  PJS - JOB REQUESTS FOR VARIABLE  ----- LINE nnnn OF nnnn
COMMAND ===> ________________________________________________ SCROLL ===> scrl

   Request-ID           Description                                    Status
ownerid.reqname   description                                         status
xxxxxxxx.xxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx
xxxxxxxx.xxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx
xxxxxxxx.xxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx
xxxxxxxx.xxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx
xxxxxxxx.xxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx
xxxxxxxx.xxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx
xxxxxxxx.xxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx
xxxxxxxx.xxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx
xxxxxxxx.xxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx
xxxxxxxx.xxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx
xxxxxxxx.xxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx
xxxxxxxx.xxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx
xxxxxxxx.xxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx
xxxxxxxx.xxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx
xxxxxxxx.xxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx
xxxxxxxx.xxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx
xxxxxxxx.xxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx
xxxxxxxx.xxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx
xxxxxxxx.xxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx
```

Figure 75: List Job Requests for Global Variable Panel (Part 1 of 3)

```
yy/mm/dd hh:mm -----  PJS - JOB REQUESTS FOR VARIABLE  ----- LINE nnnn OF nnnn
COMMAND ===> ________________________________________________ SCROLL ===> scrl

   Request-ID               JCL Data Set
ownerid.reqname   data.set.name(member)
xxxxxxxx.xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx(xxxxxxxx)
xxxxxxxx.xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx(xxxxxxxx)
xxxxxxxx.xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx(xxxxxxxx)
xxxxxxxx.xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx(xxxxxxxx)
xxxxxxxx.xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx(xxxxxxxx)
xxxxxxxx.xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx(xxxxxxxx)
xxxxxxxx.xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx(xxxxxxxx)
xxxxxxxx.xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx(xxxxxxxx)
xxxxxxxx.xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx(xxxxxxxx)
xxxxxxxx.xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx(xxxxxxxx)
xxxxxxxx.xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx(xxxxxxxx)
xxxxxxxx.xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx(xxxxxxxx)
xxxxxxxx.xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx(xxxxxxxx)
xxxxxxxx.xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx(xxxxxxxx)
xxxxxxxx.xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx(xxxxxxxx)
xxxxxxxx.xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx(xxxxxxxx)
xxxxxxxx.xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx(xxxxxxxx)
xxxxxxxx.xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx(xxxxxxxx)
xxxxxxxx.xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx(xxxxxxxx)
```

Figure 76: List Job Requests for Global Variable Panel (Part 2 of 3)

```
yy/mm/dd hh:mm -----  PJS - JOB REQUESTS FOR VARIABLE  ----- LINE nnnn OF nnnn
COMMAND ===> ________________________________________________ SCROLL ===> scrl

   Request-ID      Next Date/Time        Frequency        Events  Instdata
ownerid.reqname   mm/dd/yyyy hh:mm frequency             events   instdata
xxxxxxxx.xxxxxxxx mm/dd/yyyy hh:mm xxxxxxxxxxxxxxxxxxxxx xxxxxxxx xxxxxxxxxxxxx
xxxxxxxx.xxxxxxxx mm/dd/yyyy hh:mm xxxxxxxxxxxxxxxxxxxxx xxxxxxxx xxxxxxxxxxxxx
xxxxxxxx.xxxxxxxx mm/dd/yyyy hh:mm xxxxxxxxxxxxxxxxxxxxx xxxxxxxx xxxxxxxxxxxxx
xxxxxxxx.xxxxxxxx mm/dd/yyyy hh:mm xxxxxxxxxxxxxxxxxxxxx xxxxxxxx xxxxxxxxxxxxx
xxxxxxxx.xxxxxxxx mm/dd/yyyy hh:mm xxxxxxxxxxxxxxxxxxxxx xxxxxxxx xxxxxxxxxxxxx
xxxxxxxx.xxxxxxxx mm/dd/yyyy hh:mm xxxxxxxxxxxxxxxxxxxxx xxxxxxxx xxxxxxxxxxxxx
xxxxxxxx.xxxxxxxx mm/dd/yyyy hh:mm xxxxxxxxxxxxxxxxxxxxx xxxxxxxx xxxxxxxxxxxxx
xxxxxxxx.xxxxxxxx mm/dd/yyyy hh:mm xxxxxxxxxxxxxxxxxxxxx xxxxxxxx xxxxxxxxxxxxx
xxxxxxxx.xxxxxxxx mm/dd/yyyy hh:mm xxxxxxxxxxxxxxxxxxxxx xxxxxxxx xxxxxxxxxxxxx
xxxxxxxx.xxxxxxxx mm/dd/yyyy hh:mm xxxxxxxxxxxxxxxxxxxxx xxxxxxxx xxxxxxxxxxxxx
xxxxxxxx.xxxxxxxx mm/dd/yyyy hh:mm xxxxxxxxxxxxxxxxxxxxx xxxxxxxx xxxxxxxxxxxxx
xxxxxxxx.xxxxxxxx mm/dd/yyyy hh:mm xxxxxxxxxxxxxxxxxxxxx xxxxxxxx xxxxxxxxxxxxx
xxxxxxxx.xxxxxxxx mm/dd/yyyy hh:mm xxxxxxxxxxxxxxxxxxxxx xxxxxxxx xxxxxxxxxxxxx
xxxxxxxx.xxxxxxxx mm/dd/yyyy hh:mm xxxxxxxxxxxxxxxxxxxxx xxxxxxxx xxxxxxxxxxxxx
xxxxxxxx.xxxxxxxx mm/dd/yyyy hh:mm xxxxxxxxxxxxxxxxxxxxx xxxxxxxx xxxxxxxxxxxxx
xxxxxxxx.xxxxxxxx mm/dd/yyyy hh:mm xxxxxxxxxxxxxxxxxxxxx xxxxxxxx xxxxxxxxxxxxx
xxxxxxxx.xxxxxxxx mm/dd/yyyy hh:mm xxxxxxxxxxxxxxxxxxxxx xxxxxxxx xxxxxxxxxxxxx
xxxxxxxx.xxxxxxxx mm/dd/yyyy hh:mm xxxxxxxxxxxxxxxxxxxxx xxxxxxxx xxxxxxxxxxxxx
xxxxxxxx.xxxxxxxx mm/dd/yyyy hh:mm xxxxxxxxxxxxxxxxxxxxx xxxxxxxx xxxxxxxxxxxxx
```

Figure 77: List Job Requests for Global Variable Panel (Part 3 of 3)

For each job request, the following fields are displayed:

Request-ID
is the **Request-ID** of the job request.

Description
is the **Description** of the job request.

Status
is the **Job Request Status** of the job request. For a list of the possible values, see the section **The Job Request Status** on page 4.

JCL Data Set
is the **User's JCL Data Set** name (and member name, if a PDS is specified).

Next Run D/T
is the **Next Run Date and Time** of the job request.

Frequency
is the **Frequency** of the job request. For a list of the possible Frequency values, see the section **Specifying the Frequency** on page 6.

Events
is a summary of the **Job Request Events** specified for the job request. The following vallues may appear:

NONE	No events are specified.
POSTED	All job request events are currently posted.
xx **OF** *yy*	*yy* events are specified for the job request. Of these, *xx* are currently posted.

Instdata
is installation defined data. This is usually used for security related information, and may not be present. For detailed information on your installation data, see your PJS System Administrator.

The following commands may be entered on the **COMMAND** line:

END	returns to the **Add Global Variable** panel, the **Modify Global Variable** panel, the **Delete Global Variable** panel, the **Display Global Variable** panel, or the **List Global Variables for Owner** panel.
UP *lines*	scrolls the display up by the specified number of lines.
DOWN *lines*	scrolls the display down by the specified number of lines.
LEFT *cols*	scrolls the display left by the specified number of columns.
RIGHT *cols*	scrolls the display right by the specified number of columns.

Chapter 7: Examples

This chapter presents several common scheduling problems and solutions. The solutions presented are examples only, and other solutions may be possible.

The following examples are presented:

Example 1: How to Run a One-Time Job
shows how to use PJS to run a job once, at a later time. This example shows how to run a job with minimal specifications.

Example 2: How to Run a Daily Job
shows how to use PJS to run a job every day. This example will show how to use simple frequency options.

Example 3: How to Run a Job on the First Tuesday of Each Month
shows how to create a PJS calendar and use the calendar to schedule a job.

Example 4: How to Run One Job After Another Job Completes
shows how to use an event to schedule two jobs, one of which depends on completion of the other. Includes an example of using the **PJSPOST** job step.

Example 5: How to Set Up Several dependent Jobs
expands the previous example to include mulitple jobs with more complex dependancies.

Example 6: How to Run a Job After CICS Terminates
shows how the PJS System Administrator can set up common PJS events to enable users to submit jobs after termination of an on-line subsystem.

Example 7: How to Run a Weekly Job After a Daily Job
shows how to use the job request event **Prepost**/**Noprepost** option to solve certain types of scheduling problems.

Example 8: How to Run a Job After a Manual Operation
shows how to manually post a PJS event, using the PJS TSO commands or the PJS ISPF interface.

Example 9: How to Enable a Failed Job Request
shows what to do when a error occurs when PJS attempts to submit a job, and the job request is placed in the **ERROR** status.

Example 10: How to Modify the Submitted JCL at Submit Time
shows how to use variable substitution to modify the submitted JCL.

Example 11: How to Use a Global Variable
shows how to use a global variable to insert a common parameter into several jobs submitted by PJS.

For TSO command examples, the command entries are in **boldface**. Computer responses are in the normal font.

For ISPF panel examples, user entries are in ***italics***. High intensity fields are in **boldface**. Regular intensity fields are in the normal font.

Example 1: How to Run a One-Time Job

Problem

In this example, suppose you want to run a job at 3:00 AM tomorrow morning. Assume that the JCL for the job is in a PDS named '**TCHTST.MISC.JCL**', in the member named **MYJOB**.

TSO Solution

Use the **PJREQADD** command to add the job request as follows:

```
 READY
pjreqadd
 REQ ADD
set dataset('tchtst.misc.jcl(myjob)')
 REQ ADD
set date(*+1) time(3:00am)
 REQ ADD
end
 PJS2101I  Job request TCHTST.001 successfully added
 READY
pjreqlst id(1)
    Request-ID          Description
 TCHTST.001

                         Next Run        Window        Notify    Notify
           Status      Date    Time   Time  Option    Userid    Level
          WAIT      01/26/2005 03:00                 TCHTST    INFO

                   JCL Data Set Name                                Save
          TCHTST.MISC.JCL(MYJOB)                                   NOSAVE

             Last Submit          Status Change      Spool      Spool    Retry
            Date      Time       Date      Time      Member   Rec Count  Count
                                01/25/2005 13:04:19                     0      0

          Userid   Group
          TCHTST   WTST

                                     Start              End
              Frequency            Date    Time      Date    Time
          ONCE                  01/26/2005 03:00

 READY
```

ISPF Solution

Use the **Add Job Request** dialog to add the job request as follows:

Access the **PJS Main Menu** panel from your **ISPF Primary Option Menu** panel, or by what ever other means your installation uses. Then select option **J** to access the **PJS Job Request Menu** panel.

```
 05/01/26 14:49 -------  PJS - PERSONAL JOB SCHEDULER  ---------------- 3.1.0
 OPTION ===> J_____________________________________________________________

Select one of the following functions:

  J  JOBREQ   - Update Job Requests
  C  CALENDAR - Update Calendars
  E  EVENT    - Update Events
  V  VARIABLE - Update Variables

  T  TUTORIAL - Enter PJS/ISPF Tutorial


Select an option, then press ENTER.


(c) Copyright, Northrop Grumman, 1990, 2005.
For additional copyright information enter ABOUT on the command line.

PJS is distributed under the GNU General Public License.
For license information enter LICENSE on the command line.
```

Press **ENTER**

From the **PJS Job Request Menu** panel, select option **A** to access the **Add Job Request** panel. The **Owner-ID** defaults to the TSO User-ID. The **Request Name** is left blank (PJS will generate a Request-ID).

```
05/01/26 14:58 -------  PJS - JOB REQUEST MENU  -------------------------------
OPTION ===> A____________________________________________________________________

Select one of the following functions:

  L  LIST     - List all Job Requests for Owner
  A  ADD      - Add a new Job Request
  D  DELETE   - Delete a Job Request
  M  MODIFY   - Modify a Job Request
  S  DISPLAY  - Display a Job Request


Request-ID:
  Owner-ID     ===> TCHTST__
  Request Name ===> ________   (Required for options D, M, and S)


Enter Request-ID (if required) and make a selection, then press ENTER.
```

Press **ENTER**

From the **Add Job Request** panel, enter the **JCL** command to access the **Specify Job Request JCL Source** panel. Since we are entering as little information as possible, we will leave the **Request Description** blank.

```
05/01/26 15:06 -------  PJS - ADD JOB REQUEST  --------------------------------
COMMAND ===> JCL

Request-ID   ===> TCHTST.000
Description  ===>

Status       ===> WAIT                 Enabled      ===> YES  (Yes or No)

JCL Data Set ===>

Next Run D/T ===>
Window       ===>

Start D/T    ===>                      End D/T      ===>

Frequency    ===> ONCE

Events       ===> NONE

Commands ==> JCL - JCL Source, FREQ - Frequency, EVENT - Events,
             OPT - Submit Options, VAR - Variables,
             END - Complete Add, CANCEL - Cancel Add
```

Press **ENTER**

On the **Specify Job Request JCL Source** panel, enter the **JCL Data Set Name** and the **Member Name** in which the JCL has been placed. The **JCL Save Option** defaults to **NO**.

```
05/01/26 15:28 -------  PJS - SPECIFY JCL SOURCE  ----------------------------
COMMAND ===> ____________________________________________________________________

Request-ID   ===> TCHTST.000           Status       ===> WAIT
Description  ===>


JCL Data Set Name ===> TCHTST.MISC.JCL_____________________________
    Member Name   ===> MYJOB___

JCL Save Option   ===> NO_____       (Yes, No, or Refresh)





Commands ==> BROWSE - Browse JCL, BRDATA - Browse JCL Data Set,
             FREQ - Frequency, EVENT - Events, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
```

Press **ENTER**

After the information on the **Specify Job Request JCL Source** panel has been successfully entered, enter the **FREQ** command to access the **Specify Job Request Frequency** panel.

```
05/01/26 15:28 -------  PJS - SPECIFY JCL SOURCE  ----------------------------
COMMAND ===> FREQ_____________________________________________________________

Request-ID   ===> TCHTST.000           Status       ===> WAIT
Description  ===>


JCL Data Set Name ===> TCHTST.MISC.JCL_____________________________
    Member Name   ===> MYJOB___

JCL Save Option   ===> NO_____       (Yes, No, or Refresh)





Commands ==> BROWSE - Browse JCL, BRDATA - Browse JCL Data Set,
             FREQ - Frequency, EVENT - Events, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
```

Press **ENTER**

On the **Specify Job Request Frequency** panel, enter the **Start Date and Time**. In this case, the **Start Date** is entered as a **relative date**, to specify 'tomorrow'. The Start Time is entered as an **absolute time** to specify the specific time at which the job is to be run. The **Frequency** will default to **Once**.

```
05/01/26 15:22 -------  PJS - SPECIFY JOB FREQUENCY  -------------------------
COMMAND ===> ____________________________________________________________________

Request-ID   ===> TCHTST.000           Status       ===> WAIT
Description  ===>

Next Run D/T ===>                      Recalculate  ===> ___  (Yes or No)

Start D/T    ===> *+1_______ 03:00___  End D/T      ===> _________ ________

Frequency (choose one of the options below)
   Periodic     ===> ___ (num) _______ (units - Yr, Mo, Wk, Day, Hr, or Min)
      or
   Day of Week  ===> _ Sun  _ Mon  _ Tue  _ Wed  _ Thu  _ Fri  _ Sat
      or
   End of Month ===> Last Day - __   (days before last day of each month)
      or
   Calendar(s)  ===> _________________ _________________ _________________
      or
   Once (if none of the above is specified)

Commands ==> JCL - JCL Source, EVENT - Events, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
```

Press **ENTER**

After pressing **ENTER**, the **Next Run Date and Time** is calculated. Also note that the **relative date** we entered for the **Start Date** has been translated to an **absolute date**. After the information on the **Specify Job Request JCL Source** panel has been successfully entered, return to the **Add Job Request** panel by entering the **END** command (we will use **PF3**).

```
05/01/26 15:22 -------  PJS - SPECIFY JOB FREQUENCY  -------------------------
COMMAND ===> ____________________________________________________________________

Request-ID   ===> TCHTST.000           Status       ===> WAIT
Description  ===>

Next Run D/T ===> 01/27/2005 03:00     Recalculate  ===> ___  (Yes or No)

Start D/T    ===> 01/27/2005 03:00     End D/T      ===> _________ ________

Frequency (choose one of the options below)
   Periodic     ===> ___ (num) _______ (units - Yr, Mo, Wk, Day, Hr, or Min)
      or
   Day of Week  ===> _ Sun  _ Mon  _ Tue  _ Wed  _ Thu  _ Fri  _ Sat
      or
   End of Month ===> Last Day - __   (days before last day of each month)
      or
   Calendar(s)  ===> _______________ _______________ _______________
      or
   Once (if none of the above is specified)

Commands ==> JCL - JCL Source, EVENT - Events, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
```

Press **PF3**

We have now entered all the parameters we need for our job request. The **Add Job Request** panel has been updated to show the parameters we have entered. We now complete the add and return to the **Job Request Menu** panel by entering the **END** command (once again, we use **PF3**).

```
05/01/26 16:08 -------  PJS - ADD JOB REQUEST  --------------------------------
COMMAND ===> ____________________________________________________________________

Request-ID   ===> TCHTST.000
Description  ===> ______________________________________________________

Status       ===> WAIT                 Enabled      ===> YES  (Yes or No)

JCL Data Set ===> TCHTST.MISC.JCL(MYJOB)

Next Run D/T ===> 01/27/2005 03:00
Window       ===>

Start D/T    ===> 01/27/2005 03:00     End D/T      ===>

Frequency    ===> ONCE

Events       ===> NONE

Commands ==> JCL - JCL Source, FREQ - Frequency, EVENT - Events,
             OPT - Submit Options, VAR - Variables,
             END - Complete Add, CANCEL - Cancel Add
```

Press **PF3**

The job request add is now complete. PJS will wait until 3:00 AM tomorrow to submit the job. The **Request Name** field on the Job Request Menu panel has been updated to the Request Name generated for the job request.

```
05/01/26 16:12 -------  PJS - JOB REQUEST MENU  ------------ Job request added
OPTION ===> ____________________________________________________________________

Select one of the following functions:

  L  LIST     - List all Job Requests for Owner
  A  ADD      - Add a new Job Request
  D  DELETE   - Delete a Job Request
  M  MODIFY   - Modify a Job Request
  S  DISPLAY  - Display a Job Request


Request-ID:
  Owner-ID     ===> TCHTST__
  Request Name ===> 001_____   (Required for options D, M, and S)


Enter Request-ID (if required) and make a selection, then press ENTER.
```

Press **PF3**

Example 2: How to Run a Daily Job

Problem

In this example, suppose you want to run a job every day at 6:00 AM. Assume that the JCL for the job is in a PDS named '**TCHTST.MISC.JCL**', in the member named **MYJOB**.

TSO Solution

Use the **PJREQADD** command to add the job request as follows:

```
 READY
pjreqadd id(daily)
 REQ ADD
set desc('Example 2 - How to run a Daily Job')
 REQ ADD
set jcldsn('tchtst.misc.jcl(myjob)')
 REQ ADD
set days(1)
 REQ ADD
set strtdate(*) strttime(6:00)
 REQ ADD
end
 PJS2101I  Job request TCHTST.DAILY successfully added
 READY
pjreqlst id(daily)
    Request-ID          Description
 TCHTST.DAILY        Example 2 - How to run a Daily Job

                        Next Run        Window        Notify    Notify
            Status      Date    Time   Time  Option    Userid    Level
           WAIT      01/26/2005 06:00                TCHTST    INFO

                    JCL Data Set Name                              Save
           TCHTST.MISC.JCL(MYJOB)                                 NOSAVE

              Last Submit          Status Change      Spool      Spool    Retry
             Date      Time       Date      Time      Member   Rec Count  Count
                                01/25/2005 13:11:12                     0      0

           Userid   Group
           TCHTST   WTST

                                         Start              End
              Frequency             Date    Time      Date    Time
             1 DAYS               01/25/2005 06:00

 READY
```

ISPF Solution

Use the **Add Job Request** dialog to add the job request as follows:

Access the **PJS Main Menu** panel from your **ISPF Primary Option Menu** panel, or by what ever other means your installation uses. Then select option **J** to access the **PJS Job Request Menu** panel.

```
 05/01/26 14:49 -------  PJS - PERSONAL JOB SCHEDULER  ---------------- 3.1.0
 OPTION ===> J_____________________________________________________________

Select one of the following functions:

  J  JOBREQ   - Update Job Requests
  C  CALENDAR - Update Calendars
  E  EVENT    - Update Events
  V  VARIABLE - Update Variables

  T  TUTORIAL - Enter PJS/ISPF Tutorial


Select an option, then press ENTER.


(c) Copyright, Northrop Grumman, 1990, 2005.
For additional copyright information enter ABOUT on the command line.

PJS is distributed under the GNU General Public License.
For license information enter LICENSE on the command line.
```

Press **ENTER**

On the **PJS Job Request Menu** panel, enter the **Request Name**, then select option **A** to access the **Add Job Request** panel. The **Owner-ID** defaults to the TSO User-ID.

```
05/01/26 14:58 -------  PJS - JOB REQUEST MENU  -------------------------------
OPTION ===> A____________________________________________________________________

Select one of the following functions:

  L  LIST     - List all Job Requests for Owner
  A  ADD      - Add a new Job Request
  D  DELETE   - Delete a Job Request
  M  MODIFY   - Modify a Job Request
  S  DISPLAY  - Display a Job Request


Request-ID:
  Owner-ID     ===> TCHTST__
  Request Name ===> DAILY___   (Required for options D, M, and S)


Enter Request-ID (if required) and make a selection, then press ENTER.
```

Press **ENTER**

From the **Add Job Request** panel, enter the **Request Description**, then enter the **JCL** command to access the **Specify Job Request JCL Source** panel.

```
05/01/26 15:06 -------  PJS - ADD JOB REQUEST  --------------------------------
COMMAND ===> JCL_____________________________________________________________

Request-ID   ===> TCHTST.DAILY
Description  ===> Example 2 - How to Run a Daily Job_________________

Status       ===> WAIT                 Enabled      ===> YES  (Yes or No)

JCL Data Set ===>

Next Run D/T ===>
Window       ===>

Start D/T    ===>                      End D/T      ===>

Frequency    ===> ONCE

Events       ===> NONE

Commands ==> JCL - JCL Source, FREQ - Frequency, EVENT - Events,
             OPT - Submit Options, VAR - Variables,
             END - Complete Add, CANCEL - Cancel Add
```

Press **ENTER**

On the **Specify Job Request JCL Source** panel, enter the **JCL Data Set Name** and the **Member Name** in which the JCL has been placed. The **JCL Save Option** defaults to **NO**.

```
05/01/26 15:28 -------  PJS - SPECIFY JCL SOURCE  ----------------------------
COMMAND ===> ____________________________________________________________________

Request-ID   ===> TCHTST.DAILY         Status       ===> WAIT
Description  ===> Example 2 - How to Run a Daily Job


JCL Data Set Name ===> TCHTST.MISC.JCL_____________________________
    Member Name   ===> MYJOB___

JCL Save Option   ===> NO_____       (Yes, No, or Refresh)





Commands ==> BROWSE - Browse JCL, BRDATA - Browse JCL Data Set,
             FREQ - Frequency, EVENT - Events, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
```

Press **ENTER**

After the information on the **Specify Job Request JCL Source** panel has been successfully entered, enter the **FREQ** command to access the **Specify Job Request Frequency** panel.

```
05/01/26 15:28 -------  PJS - SPECIFY JCL SOURCE  ----------------------------
COMMAND ===> FREQ_____________________________________________________________

Request-ID   ===> TCHTST.DAILY         Status       ===> WAIT
Description  ===> Example 2 - How to Run a Daily Job


JCL Data Set Name ===> TCHTST.MISC.JCL_____________________________
    Member Name   ===> MYJOB___

JCL Save Option   ===> NO_____      (Yes, No, or Refresh)





Commands ==> BROWSE - Browse JCL, BRDATA - Browse JCL Data Set,
             FREQ - Frequency, EVENT - Events, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
```

Press **ENTER**

On the **Specify Job Request Frequency** panel, enter the **Start Date and Time** and the **Frequency**. In this case, the **Start Date** is entered as a **relative date**, to specify 'today'. The Start Time is entered as an **absolute time** to specify the specific time at which the job is to be run. The **Frequency** is entered as a **Periodic** frequency by entering **1** in the number field, and **DAYS** in the units field.

```
05/01/26 15:22 -------  PJS - SPECIFY JOB FREQUENCY  --------------------------
COMMAND ===> ____________________________________________________________________

Request-ID   ===> TCHTST.DAILY         Status       ===> WAIT
Description  ===> Example 2 - How to Run a Daily Job

Next Run D/T ===>                      Recalculate  ===> ___  (Yes or No)

Start D/T    ===> *________ 06:00___   End D/T      ===> _________ ________

Frequency (choose one of the options below)
   Periodic     ===> 1__ (num) DAYS___ (units - Yr, Mo, Wk, Day, Hr, or Min)
      or
   Day of Week  ===> _ Sun  _ Mon  _ Tue  _ Wed  _ Thu  _ Fri  _ Sat
      or
   End of Month ===> Last Day - __   (days before last day of each month)
      or
   Calendar(s)  ===> _______________ _______________ _______________
      or
   Once (if none of the above is specified)

Commands ==> JCL - JCL Source, EVENT - Events, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
```

Press **ENTER**

After pressing **ENTER**, the **Next Run Date and Time** is calculated. Even though the start date was enter as 'today', the next run date and time is set to 'tomorrow', since the start time for today is already past. Also note that the **relative date** we entered for the **Start Date** has been translated to an **absolute date**. After the information on the **Specify Job Request JCL Source** panel has been successfully entered, return to the **Add Job Request** panel by entering the **END** command (we will use **PF3**).

```
05/01/26 15:22 -------  PJS - SPECIFY JOB FREQUENCY  --------------------------
COMMAND ===> ____________________________________________________________________

Request-ID   ===> TCHTST.DAILY         Status       ===> WAIT
Description  ===> Example 2 - How to Run a Daily Job

Next Run D/T ===> 01/27/2005 06:00     Recalculate  ===> ___  (Yes or No)

Start D/T    ===> 01/26/2005 06:00     End D/T      ===> __________ ________

Frequency (choose one of the options below)
   Periodic     ===> 001 (num) DAYS___ (units - Yr, Mo, Wk, Day, Hr, or Min)
      or
   Day of Week  ===> _ Sun  _ Mon  _ Tue  _ Wed  _ Thu  _ Fri  _ Sat
      or
   End of Month ===> Last Day - __   (days before last day of each month)
      or
   Calendar(s)  ===> _________________ _________________ _________________
      or
   Once (if none of the above is specified)

Commands ==> JCL - JCL Source, EVENT - Events, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
```

Press **PF3**

We have now entered all the parameters we need for our job request. The **Add Job Request** panel has been updated to show the parameters we have entered. We now complete the add and return to the **Job Request Menu** panel by entering the **END** command (once again, we use **PF3**).

```
05/01/26 16:08 -------  PJS - ADD JOB REQUEST  --------------------------------
COMMAND ===> ____________________________________________________________________

Request-ID   ===> TCHTST.DAILY
Description  ===> Example 2 - How to Run a Daily Job_________________

Status       ===> WAIT                 Enabled      ===> YES  (Yes or No)

JCL Data Set ===> TCHTST.MISC.JCL(MYJOB)

Next Run D/T ===> 01/27/2005 06:00
Window       ===>

Start D/T    ===> 01/26/2005 06:00     End D/T      ===>

Frequency    ===> 001 DAYS

Events       ===> NONE

Commands ==> JCL - JCL Source, FREQ - Frequency, EVENT - Events,
             OPT - Submit Options, VAR - Variables,
             END - Complete Add, CANCEL - Cancel Add
```

Press **PF3**

The job request add is now complete. PJS will submit the job every morning at 6:00 AM.

```
05/01/26 16:12 -------  PJS - JOB REQUEST MENU  ------------ Job request added
OPTION ===> ____________________________________________________________________

Select one of the following functions:

  L  LIST     - List all Job Requests for Owner
  A  ADD      - Add a new Job Request
  D  DELETE   - Delete a Job Request
  M  MODIFY   - Modify a Job Request
  S  DISPLAY  - Display a Job Request


Request-ID:
  Owner-ID     ===> TCHTST__
  Request Name ===> DAILY___   (Required for options D, M, and S)


Enter Request-ID (if required) and make a selection, then press ENTER.
```

Press **PF3**

Example 3: How to Run a Job on the First Tuesday of Each Month

Problem

In this example, suppose you want to run a job on the first tuesday of each month at 6:00 PM. Assume that the JCL for the job is in a PDS named '**TCHTST.MISC.JCL**', in the member named **MYJOB**.

To schedule this job, we will first define a calendar to select the first tuesday of each month for the next three years. If we need the job to continue after this time, will will need to modify the calendar to add more dates.

TSO Solution

Use the **PJCALADD** command to add the job request as follows:

```
 READY
pjcaladd firsttue desc('First Tuesday of each month')
                  dates(2/1/05 3/1/05 4/5/05 5/3/05 6/7/05 7/5/05
                        8/2/05 9/6/05 10/4/05 11/1/05 12/6/05 1/3/06
                        2/7/06 3/7/06 4/4/06 5/2/06 6/6/06 7/4/06
                        8/1/06 9/5/06 10/3/06 11/7/06 12/5/06 1/2/07
                        2/6/07 3/6/07 4/3/07 5/1/07 6/5/07 7/3/07
                        8/7/07 9/4/07 10/2/07 11/6/07 12/4/07)
    Calendar-ID          Description
 TCHTST.FIRSTTUE    First Tuesday of each month

                       Dates
                    02/01/2005  03/01/2005  04/05/2005  05/03/2005  06/07/2005
                    07/05/2005  08/02/2005  09/06/2005  10/04/2005  11/01/2005
                    12/06/2005  01/03/2006  02/07/2006  03/07/2006  04/04/2006
                    05/02/2006  06/06/2006  07/04/2006  08/01/2006  09/05/2006
                    10/03/2006  11/07/2006  12/05/2006  01/02/2007  02/06/2007
                    03/06/2007  04/03/2007  05/01/2007  06/05/2007  07/03/2007
                    08/07/2007  09/04/2007  10/02/2007  11/06/2007  12/04/2007

 PJS2201I  Calendar TCHTST.FIRSTTUE  successfully added
 READY
```

Next, use the **PJREQADD** command to add the job request as follows:

```
 READY
pjreqadd id(example3)
 REQ ADD
set desc('Example 3 - How to Run a Job Using a Calendar')
 REQ ADD
set jcldsn('tchtst.misc.jcl(myjob)')
 REQ ADD
set strtdate(*) strttime(6:00pm) calendar(firsttue)
 REQ ADD
end
 PJS2101I  Job request TCHTST.EXAMPLE3 successfully added
 READY
pjreqlst id(example3)
    Request-ID          Description
 TCHTST.EXAMPLE3    Example 3 - How to Run a Job Using a Calendar

                         Next Run        Window        Notify    Notify
           Status       Date    Time   Time  Option    Userid    Level
          WAIT      02/01/2005 18:00                 TCHTST    INFO

                   JCL Data Set Name                                Save
          TCHTST.MISC.JCL(MYJOB)                                   NOSAVE

             Last Submit          Status Change      Spool      Spool    Retry
            Date      Time       Date      Time      Member   Rec Count  Count
                                01/25/2005 13:35:35                    0      0

          Userid   Group
          TCHTST   WTST

                                     Start              End
              Frequency            Date    Time      Date    Time
          CAL-TCHTST.FIRSTTUE    01/25/2005 18:00

 READY
```

ISPF Solution

Use the **Add Calendar** dialog to add the calendar as follows:

Access the **PJS Main Menu** panel from your **ISPF Primary Option Menu** panel, or by what ever other means your installation uses. Then select option **C** to access the **PJS Calendar Menu** panel.

```
 05/01/26 14:49 -------  PJS - PERSONAL JOB SCHEDULER  ---------------- 3.1.0
 OPTION ===> C_____________________________________________________________

Select one of the following functions:

  J  JOBREQ   - Update Job Requests
  C  CALENDAR - Update Calendars
  E  EVENT    - Update Events
  V  VARIABLE - Update Variables

  T  TUTORIAL - Enter PJS/ISPF Tutorial


Select an option, then press ENTER.


(c) Copyright, Northrop Grumman, 1990, 2005.
For additional copyright information enter ABOUT on the command line.

PJS is distributed under the GNU General Public License.
For license information enter LICENSE on the command line.
```

Press **ENTER**

On the **PJS Calendar Menu** panel, enter the **Calendar Name**, then select option **A** to access the **Add Calendar** panel. The **Owner-ID** defaults to the TSO User-ID.

```
05/01/26 14:47 -------  PJS - CALENDAR MENU  ----------------------------------
OPTION ===> A____________________________________________________________________

Select one of the following functions:

  L  LIST     - List all Calendars for Owner-ID
  A  ADD      - Add a new Calendar
  D  DELETE   - Delete a Calendar
  M  MODIFY   - Modify a Calendar
  S  DISPLAY  - Display a Calendar


Calendar-ID:
  Owner-ID      ===> TCHTST__
  Calendar Name ===> FIRSTTUE       (Required for options A, D, M, and S)


Enter Calendar-ID (if required) and make a selection, then press ENTER.
```

Press **ENTER**

On the **Add Calendar** panel, enter the **Calendar Description**, then enter the **Month** and **Year** of the first month in which dates are to be selected (the current month is initially displayed). Since the calendar initially has no dates selected, a warning message is displayed in the upper right hand corner of the panel.

```
05/01/26 14:55 -------  PJS - ADD CALENDAR  ------------ Calendar has no dates
COMMAND ===> ____________________________________________________________________

Calendar-ID  ===> TCHTST.FIRSTTUE
Description  ===> Example 3 - How to Run a Job Using a Calendar_____

               Month FEBRUARY_                Year 2005

                                                    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  _ 29
             _ 30  _ 31

Enter the 3-letter month and year above to select month, or
Enter scroll commands (or use PF keys) to select next/previous month

Enter a non-blank character before each date to be selected
Enter a Space before each date to be excluded

Commands ==> REQ - Display Job Requests for Calendar,
             END - Complete Add, CANCEL - Cancel Add
```

Press **ENTER**

The **Add Calendar** panel will be redisplayed with the selected month. You can now select the dates required for this month. Enter a non-blank character in the unprotected field immediately before the date to be selected (01).

```
05/01/26 11:08 -------  PJS - ADD CALENDAR  ------------ Calendar has no dates
COMMAND ===> ____________________________________________________________________

Calendar-ID  ===> TCHTST.FIRSTTUE
Description  ===> Example 3 - How to Run a Job Using a Calendar_____

               Month FEBRUARY_                Year 2005

               Sun   Mon   Tue   Wed   Thu   Fri   Sat
                           x 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

Enter the 3-letter month and year above to select month, or
Enter scroll commands (or use PF keys) to select next/previous month

Enter a non-blank character before each date to be selected
Enter a Space before each date to be excluded

Commands ==> REQ - Display Job Requests for Calendar,
             END - Complete Add, CANCEL - Cancel Add
```

Press **ENTER**

The selected dates are now highlighted, and the unprotected field now contains an asterisk ('*'). Also, the warning message has been removed. The next dates to be selected are in the next month. Display the next month by entering the **DOWN** command (we will use **PF8**).

```
05/01/26 11:08 -------  PJS - ADD CALENDAR  ----------------------------------
COMMAND ===> ______________________________________________________________________

Calendar-ID  ===> TCHTST.FIRSTTUE
Description  ===> Example 3 - How to Run a Job Using a Calendar_____

               Month FEBRUARY_                Year 2005

               Sun   Mon   Tue   Wed   Thu   Fri   Sat
                           * 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

Enter the 3-letter month and year above to select month, or
Enter scroll commands (or use PF keys) to select next/previous month

Enter a non-blank character before each date to be selected
Enter a Space before each date to be excluded

Commands ==> REQ - Display Job Requests for Calendar,
             END - Complete Add, CANCEL - Cancel Add
```

Press **PF8**

The **Add Calendar** panel is now redisplayed with the next month (February). Select the dates for this month.

```
05/01/26 11:08 -------  PJS - ADD CALENDAR  -----------------------------------
COMMAND ===> ____________________________________________________________________

Calendar-ID  ===> TCHTST.FIRSTTUE
Description  ===> Example 3 - How to Run a Job Using a Calendar_____

               Month MARCH____                Year 2005

               Sun   Mon   Tue   Wed   Thu   Fri   Sat
                           x 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  _ 29  _ 30  _ 31

Enter the 3-letter month and year above to select month, or
Enter scroll commands (or use PF keys) to select next/previous month

Enter a non-blank character before each date to be selected
Enter a Space before each date to be excluded

Commands ==> REQ - Display Job Requests for Calendar,
             END - Complete Add, CANCEL - Cancel Add
```

Press **ENTER**

Continue scrolling to the next month and selecting dates until all dates have been selected. For brievity, each month is not shown here. We will skip to the last month.

After selecting all of the desired dates, we will complete the add and return to the **Calendar Menu** panel by entering the **END** command (we will use **PF3**).

```
05/01/26 11:08 -------  PJS - ADD CALENDAR  -----------------------------------
COMMAND ===> ______________________________________________________________________

Calendar-ID  ===> TCHTST.FIRSTTUE
Description  ===> Example 3 - How to Run a Job Using a Calendar_____

               Month DECEMBER_                Year 2007

               Sun   Mon   Tue   Wed   Thu   Fri   Sat
                                                   _ 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  _ 29
              _ 30  _ 31

Enter the 3-letter month and year above to select month, or
Enter scroll commands (or use PF keys) to select next/previous month

Enter a non-blank character before each date to be selected
Enter a Space before each date to be excluded

Commands ==> REQ - Display Job Requests for Calendar,
             END - Complete Add, CANCEL - Cancel Add
```

Press **PF3**

The calendar add is now complete, and the calendar can be used in a job request. Return to the **PJS Main Menu** panel by entering the **END** command (once again, we use **PF3**).

```
05/01/26 14:47 -------  PJS - CALENDAR MENU  ------------------ Calendar added
OPTION ===> ____________________________________________________________________

Select one of the following functions:

  L  LIST     - List all Calendars for Owner-ID
  A  ADD      - Add a new Calendar
  D  DELETE   - Delete a Calendar
  M  MODIFY   - Modify a Calendar
  S  DISPLAY  - Display a Calendar


Calendar-ID:
  Owner-ID      ===> TCHTST__
  Calendar Name ===> FIRSTTUE       (Required for options A, D, M, and S)


Enter Calendar-ID (if required) and make a selection, then press ENTER.
```

Press **PF3**

Next, use the **Add Job Request** dialog to add the job request as follows:

From the **PJS Main Menu** panel, select option **J** to access the **PJS Job Request Menu** panel.

```
 05/01/26 14:49 -------  PJS - PERSONAL JOB SCHEDULER  ---------------- 3.1.0
 OPTION ===> J_____________________________________________________________________

Select one of the following functions:

  J  JOBREQ   - Update Job Requests
  C  CALENDAR - Update Calendars
  E  EVENT    - Update Events
  V  VARIABLE - Update Variables

  T  TUTORIAL - Enter PJS/ISPF Tutorial


Select an option, then press ENTER.


(c) Copyright, Northrop Grumman, 1990, 2005.
For additional copyright information enter ABOUT on the command line.

PJS is distributed under the GNU General Public License.
For license information enter LICENSE on the command line.
```

Press **ENTER**

On the **PJS Job Request Menu** panel, enter the **Request Name**, then select option **A** to access the **Add Job Request** panel. The **Owner-ID** defaults to the TSO User-ID.

```
05/01/26 14:58 -------  PJS - JOB REQUEST MENU  -------------------------------
OPTION ===> A

Select one of the following functions:

  L  LIST     - List all Job Requests for Owner
  A  ADD      - Add a new Job Request
  D  DELETE   - Delete a Job Request
  M  MODIFY   - Modify a Job Request
  S  DISPLAY  - Display a Job Request


Request-ID:
  Owner-ID     ===> TCHTST
  Request Name ===> EXAMPLE3   (Required for options D, M, and S)


Enter Request-ID (if required) and make a selection, then press ENTER.
```

Press **ENTER**

From the **Add Job Request** panel, enter the **Request Description**, then enter the **JCL** command to access the **Specify Job Request JCL Source** panel.

```
05/01/26 15:06 -------  PJS - ADD JOB REQUEST  --------------------------------
COMMAND ===> JCL_____________________________________________________________

Request-ID   ===> TCHTST.EXAMPLE3
Description  ===> Example 3 - How to Run a Job Using a Calendar_____

Status       ===> WAIT                 Enabled      ===> YES  (Yes or No)

JCL Data Set ===>

Next Run D/T ===>
Window       ===>

Start D/T    ===>                      End D/T      ===>

Frequency    ===> ONCE

Events       ===> NONE

Commands ==> JCL - JCL Source, FREQ - Frequency, EVENT - Events,
             OPT - Submit Options, VAR - Variables,
             END - Complete Add, CANCEL - Cancel Add
```

Press **ENTER**

On the **Specify Job Request JCL Source** panel, enter the **JCL Data Set Name** and the **Member Name** in which the JCL has been placed. The **JCL Save Option** defaults to **NO**.

```
05/01/26 15:28 -------  PJS - SPECIFY JCL SOURCE  ----------------------------
COMMAND ===> ____________________________________________________________________

Request-ID   ===> TCHTST.EXAMPLE3      Status       ===> WAIT
Description  ===>  Example 3 - How to Run a Job Using a Calendar


JCL Data Set Name ===> TCHTST.MISC.JCL_____________________________
    Member Name   ===> MYJOB___

JCL Save Option   ===> NO_____      (Yes, No, or Refresh)





Commands ==> BROWSE - Browse JCL, BRDATA - Browse JCL Data Set,
             FREQ - Frequency, EVENT - Events, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
```

Press **ENTER**

After the information on the **Specify Job Request JCL Source** panel has been successfully entered, enter the **FREQ** command to access the **Specify Job Request Frequency** panel.

```
05/01/26 15:28 -------  PJS - SPECIFY JCL SOURCE  ----------------------------
COMMAND ===> FREQ_____________________________________________________________

Request-ID   ===> TCHTST.EXAMPLE3      Status       ===> WAIT
Description  ===> Example 3 - How to Run a Job Using a Calendar


JCL Data Set Name ===> TCHTST.MISC.JCL_____________________________
    Member Name   ===> MYJOB___

JCL Save Option   ===> NO_____      (Yes, No, or Refresh)





Commands ==> BROWSE - Browse JCL, BRDATA - Browse JCL Data Set,
             FREQ - Frequency, EVENT - Events, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
```

Press **ENTER**

On the **Specify Job Request Frequency** panel, enter the **Start Date and Time** and the **Frequency**. In this case, the **Start Date** is entered as a **relative date**, to specify 'today'. The Start Time is entered as an **absolute time** to specify the specific time at which the job is to be run. The **Frequency** is entered as a **Calendar** frequency by entering the Calendar-ID of the calendar to be used.

```
05/01/26 15:22 -------  PJS - SPECIFY JOB FREQUENCY  -------------------------
COMMAND ===> ______________________________________________________________________

Request-ID   ===> TCHTST.EXAMPLE3      Status       ===> WAIT
Description  ===> Example 3 - How to Run a Job Using a Calendar

Next Run D/T ===>                      Recalculate  ===> ___  (Yes or No)

Start D/T    ===> *________ 06:00 pm  End D/T      ===> _________ ________

Frequency (choose one of the options below)
   Periodic     ===> ___ (num) _______ (units - Yr, Mo, Wk, Day, Hr, or Min)
      or
   Day of Week  ===> _ Sun  _ Mon  _ Tue  _ Wed  _ Thu  _ Fri  _ Sat
      or
   End of Month ===> Last Day - __   (days before last day of each month)
      or
   Calendar(s)  ===> firsttue________ _________________ _________________
      or
   Once (if none of the above is specified)

Commands ==> JCL - JCL Source, EVENT - Events, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
```

Press **ENTER**

After pressing **ENTER**, the **Next Run Date and Time** is calculated. Even though the start date was enter as 'today', the next run date is set to the next date selected by the calendar. Also note that the **relative date** we entered for the **Start Date** has been translated to an **absolute date**. After the information on the **Specify Job Request JCL Source** panel has been successfully entered, return to the **Add Job Request** panel by entering the **END** command (we will use **PF3**).

```
05/01/26 15:22 -------  PJS - SPECIFY JOB FREQUENCY  -------------------------
COMMAND ===> ____________________________________________________________________

Request-ID   ===> TCHTST.EXAMPLE3      Status       ===> WAIT
Description  ===> Example 3 - How to Run a Job Using a Calendar

Next Run D/T ===> 02/01/2005 18:00     Recalculate  ===> ___  (Yes or No)

Start D/T    ===> 01/26/2005 18:00     End D/T      ===> _________ ________

Frequency (choose one of the options below)
   Periodic     ===> ___ (num) _______ (units - Yr, Mo, Wk, Day, Hr, or Min)
      or
   Day of Week  ===> _ Sun  _ Mon  _ Tue  _ Wed  _ Thu  _ Fri  _ Sat
      or
   End of Month ===> Last Day - __   (days before last day of each month)
      or
   Calendar(s)  ===> TCHTST.FIRSTTUE__ _________________ _________________
      or
   Once (if none of the above is specified)

Commands ==> JCL - JCL Source, EVENT - Events, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
```

Press **PF3**

We have now entered all the parameters we need for our job request. The **Add Job Request** panel has been updated to show the parameters we have entered. We now complete the add and return to the **Job Request Menu** panel by entering the **END** command (once again, we use **PF3**).

```
05/01/26 16:08 -------  PJS - ADD JOB REQUEST  --------------------------------
COMMAND ===> ____________________________________________________________________

Request-ID   ===> TCHTST.EXAMPLE3
Description  ===> Example 3 - How to Run a Job Using a Calendar_____

Status       ===> WAIT                 Enabled      ===> YES  (Yes or No)

JCL Data Set ===> TCHTST.MISC.JCL(MYJOB)

Next Run D/T ===> 02/01/2005 18:00
Window       ===>

Start D/T    ===> 01/26/2005 18:00     End D/T      ===>

Frequency    ===> CAL-TCHTST.FIRSTTUE

Events       ===> NONE

Commands ==> JCL - JCL Source, FREQ - Frequency, EVENT - Events,
             OPT - Submit Options, VAR - Variables,
             END - Complete Add, CANCEL - Cancel Add
```

Press **PF3**

The job request add is now complete. PJS will submit the job every morning at 6:00 AM.

```
05/01/26 16:12 -------  PJS - JOB REQUEST MENU  ------------ Job request added
OPTION ===> ____________________________________________________________________

Select one of the following functions:

  L  LIST     - List all Job Requests for Owner
  A  ADD      - Add a new Job Request
  D  DELETE   - Delete a Job Request
  M  MODIFY   - Modify a Job Request
  S  DISPLAY  - Display a Job Request


Request-ID:
  Owner-ID     ===> TCHTST__
  Request Name ===> EXAMPLE3   (Required for options D, M, and S)


Enter Request-ID (if required) and make a selection, then press ENTER.
```

Press **PF3**

Example 4: How to Run One Job After Another Job Completes

Problem

In this example, suppose you want to run two jobs. The first job is to be submitted at 6:00 PM today. The second job is to be submitted after the first successfully completes. Assume that the JCL for the jobs is in a PDS named '**TCHTST.MISC.JCL**'. The first job is in the member named **JOB1**. The second job is in the member named **JOB2**.

To schedule these jobs, we will use an event that we will name **JOB1END**. The JCL for JOB1 will be modified to post this event as the last step, if the previous steps have completed successfully. JOB1 will be scheduled with a simple job request, as shown in Example 1. JOB2 will be scheduled with a job request that specifies a run time the same as JOB1, but will also specify that the event must be posted before the job is run.

Before defining the job requests to PJS, first place the JCL in the specified data set and members. Modify JOB1 to include the PJSPOST step at the end of the job as follows:

```
//JOB1     JOB
//  existing JCL statements
//            .
//            .
//            .
//PJSPOST EXEC PGM=PJSPOST,PARM='TCHTST.JOB1END',
//             COND=(0,NE)
```

The **PARM** parameter specifies the name of the event to be posted. The **COND** parameter is used to post the event only if all previous job steps completed with zero return codes.

TSO Solution

Use the **PJREQADD** command to add the job request for JOB1 as follows:

```
 READY
pjreqadd id(ex4job1)
 REQ ADD
set desc('Example 4 - Job 1 - Using Events')
 REQ ADD
set jcldsn('tchtst.misc.jcl(job1)')
 REQ ADD
set strtdate(*) strttime(6:00pm)
 REQ ADD
end
 PJS2101I  Job request TCHTST.EX4JOB1 successfully added
 READY
pjreqlst id(ex4job1)
 TCHTST.EX4JOB1     Example 4 - Job 1 - Using Events

                         Next Run        Window        Notify    Notify
           Status      Date    Time   Time  Option    Userid    Level
          WAIT      01/25/2005 18:00                 TCHTST    INFO

                    JCL Data Set Name                              Save
          TCHTST.MISC.JCL(JOB1)                                   NOSAVE

             Last Submit          Status Change      Spool      Spool    Retry
            Date      Time       Date      Time      Member   Rec Count  Count
                                01/25/2005 13:45:27                     0      0

          Userid   Group
          TCHTST   WTST

                                     Start              End
              Frequency            Date    Time      Date    Time
          ONCE                  01/25/2005 18:00

 READY
```

Next, use the **PJREQADD** command to add the job request for JOB2 as follows:

```
 READY
pjreqadd id(ex4job2)
 REQ ADD
set desc('Example 4 - Job 2 - Using Events')
 REQ ADD
set jcldsn('tchtst.misc.jcl(job2)')
 REQ ADD
set strtdate(*) strttime(6:00pm)
 REQ ADD
event add job1end
 REQ ADD
end
 PJS2101I  Job request TCHTST.EX4JOB2 successfully added
 READY
pjreqlst id(ex4job2)
 TCHTST.EX4JOB2     Example 4 - Job 2 - Using Events

                        Next Run        Window        Notify    Notify
           Status      Date    Time   Time  Option    Userid    Level
          WAIT      01/25/2005 18:00                 TCHTST    INFO

                    JCL Data Set Name                              Save
          TCHTST.MISC.JCL(JOB2)                                   NOSAVE

             Last Submit          Status Change      Spool      Spool    Retry
            Date      Time       Date      Time      Member   Rec Count  Count
                                01/25/2005 14:07:42                    0      0

          Userid   Group
          TCHTST   WTST

                                     Start              End
             Frequency             Date    Time      Date    Time
          ONCE                   01/25/2005 18:00

                            Pre-        Posted
             Event-ID       Post    Date      Time
          TCHTST.JOB1END    YES

 READY
```

ISPF Solution

Use the **Add Job Request** dialog to add the job request for the first job as follows:

Access the **PJS Main Menu** panel from your **ISPF Primary Option Menu** panel, or by what ever other means your installation uses. Then select option **J** to access the **PJS Job Request Menu** panel.

```
 05/01/26 14:49 -------  PJS - PERSONAL JOB SCHEDULER  ---------------- 3.1.0
 OPTION ===> J

Select one of the following functions:

  J  JOBREQ   - Update Job Requests
  C  CALENDAR - Update Calendars
  E  EVENT    - Update Events
  V  VARIABLE - Update Variables

  T  TUTORIAL - Enter PJS/ISPF Tutorial


Select an option, then press ENTER.


(c) Copyright, Northrop Grumman, 1990, 2005.
For additional copyright information enter ABOUT on the command line.

PJS is distributed under the GNU General Public License.
For license information enter LICENSE on the command line.
```

Press **ENTER**

On the **PJS Job Request Menu** panel, enter the **Request Name**, then select option **A** to access the **Add Job Request** panel. The **Owner-ID** defaults to the TSO User-ID.

```
05/01/26 14:58 -------  PJS - JOB REQUEST MENU  -------------------------------
OPTION ===> A____________________________________________________________________

Select one of the following functions:

  L  LIST     - List all Job Requests for Owner
  A  ADD      - Add a new Job Request
  D  DELETE   - Delete a Job Request
  M  MODIFY   - Modify a Job Request
  S  DISPLAY  - Display a Job Request


Request-ID:
  Owner-ID     ===> TCHTST__
  Request Name ===> EX4JOB1_   (Required for options D, M, and S)


Enter Request-ID (if required) and make a selection, then press ENTER.
```

Press **ENTER**

From the **Add Job Request** panel, enter the **Request Description**, then enter the **JCL** command to access the **Specify Job Request JCL Source** panel.

```
05/01/26 15:06 -------  PJS - ADD JOB REQUEST  --------------------------------
COMMAND ===> JCL_____________________________________________________________

Request-ID   ===> TCHTST.EX4JOB1
Description  ===> Example 4 - Job 1 - Using Events_________________

Status       ===> WAIT                 Enabled      ===> YES  (Yes or No)

JCL Data Set ===>

Next Run D/T ===>
Window       ===>

Start D/T    ===>                      End D/T      ===>

Frequency    ===> ONCE

Events       ===> NONE

Commands ==> JCL - JCL Source, FREQ - Frequency, EVENT - Events,
             OPT - Submit Options, VAR - Variables,
             END - Complete Add, CANCEL - Cancel Add
```

Press **ENTER**

On the **Specify Job Request JCL Source** panel, enter the **JCL Data Set Name** and the **Member Name** in which the JCL has been placed. The **JCL Save Option** defaults to **NO**.

```
05/01/26 15:28 -------  PJS - SPECIFY JCL SOURCE  ----------------------------
COMMAND ===> ____________________________________________________________________

Request-ID   ===> TCHTST.EX4JOB1       Status       ===> WAIT
Description  ===>  Example 4 - Job 1 - Using Events


JCL Data Set Name ===> TCHTST.MISC.JCL_____________________________
    Member Name   ===> JOB1____

JCL Save Option   ===> NO_____      (Yes, No, or Refresh)





Commands ==> BROWSE - Browse JCL, BRDATA - Browse JCL Data Set,
             FREQ - Frequency, EVENT - Events, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
```

Press **ENTER**

After the information on the **Specify Job Request JCL Source** panel has been successfully entered, enter the **FREQ** command to access the **Specify Job Request Frequency** panel.

```
05/01/26 15:28 -------  PJS - SPECIFY JCL SOURCE  ----------------------------
COMMAND ===> FREQ_____________________________________________________________

Request-ID   ===> TCHTST.EX4JOB1       Status       ===> WAIT
Description  ===> Example 4 - Job 1 - Using Events


JCL Data Set Name ===> TCHTST.MISC.JCL_____________________________
    Member Name   ===> JOB1____

JCL Save Option   ===> NO_____      (Yes, No, or Refresh)





Commands ==> BROWSE - Browse JCL, BRDATA - Browse JCL Data Set,
             FREQ - Frequency, EVENT - Events, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
```

Press **ENTER**

On the **Specify Job Request Frequency** panel, enter the **Start Date and Time**. In this case, the **Start Date** is entered as a **relative date**, to specify 'today'. The Start Time is entered as an **absolute time** to specify the specific time at which the job is to be run. The **Frequency** will default to **Once**.

```
05/01/26 15:22 -------  PJS - SPECIFY JOB FREQUENCY  --------------------------
COMMAND ===> ____________________________________________________________________

Request-ID   ===> TCHTST.EX4JOB1       Status       ===> WAIT
Description  ===> Example 4 - Job 1 - Using Events

Next Run D/T ===>                      Recalculate  ===> ___  (Yes or No)

Start D/T    ===> *________ 06:00 pm   End D/T      ===> _________ ________

Frequency (choose one of the options below)
   Periodic     ===> ___ (num) _______ (units - Yr, Mo, Wk, Day, Hr, or Min)
      or
   Day of Week  ===> _ Sun  _ Mon  _ Tue  _ Wed  _ Thu  _ Fri  _ Sat
      or
   End of Month ===> Last Day - __   (days before last day of each month)
      or
   Calendar(s)  ===> _______________ _______________ _______________
      or
   Once (if none of the above is specified)

Commands ==> JCL - JCL Source, EVENT - Events, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
```

Press **ENTER**

After pressing **ENTER**, the **Next Run Date and Time** is calculated. Also note that the **relative date** we entered for the **Start Date** has been translated to an **absolute date**. After the information on the **Specify Job Request JCL Source** panel has been successfully entered, return to the **Add Job Request** panel by entering the **END** command (we will use **PF3**).

```
05/01/26 15:22 -------  PJS - SPECIFY JOB FREQUENCY  --------------------------
COMMAND ===> ____________________________________________________________________

Request-ID   ===> TCHTST.EX4JOB1       Status       ===> WAIT
Description  ===> Example 4 - Job 1 - Using Events

Next Run D/T ===> 01/26/2005 18:00     Recalculate  ===> ___  (Yes or No)

Start D/T    ===> 01/26/2005 18:00     End D/T      ===> _________ ________

Frequency (choose one of the options below)
   Periodic     ===> ___ (num) _______ (units - Yr, Mo, Wk, Day, Hr, or Min)
      or
   Day of Week  ===> _ Sun  _ Mon  _ Tue  _ Wed  _ Thu  _ Fri  _ Sat
      or
   End of Month ===> Last Day - __   (days before last day of each month)
      or
   Calendar(s)  ===> _________________ _________________ _________________
      or
   Once (if none of the above is specified)

Commands ==> JCL - JCL Source, EVENT - Events, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
```

Press **PF3**

We have now entered all the parameters we need for this job request. The **Add Job Request** panel has been updated to show the parameters we have entered. We now complete the add and return to the **Job Request Menu** panel by entering the **END** command (once again, we use **PF3**).

```
05/01/26 16:08 -------  PJS - ADD JOB REQUEST  --------------------------------
COMMAND ===> ____________________________________________________________________

Request-ID   ===> TCHTST.EX4JOB1
Description  ===> Example 4 - Job 1 - Using Events_________________

Status       ===> WAIT                 Enabled      ===> YES  (Yes or No)

JCL Data Set ===> TCHTST.MISC.JCL(JOB1)

Next Run D/T ===> 01/26/2005 18:00
Window       ===>

Start D/T    ===> 01/26/2005 18:00     End D/T      ===>

Frequency    ===> ONCE

Events       ===> NONE

Commands ==> JCL - JCL Source, FREQ - Frequency, EVENT - Events,
             OPT - Submit Options, VAR - Variables,
             END - Complete Add, CANCEL - Cancel Add
```

Press **PF3**

The first job request add is now complete. PJS will submit this at 6:00 PM today.

Next, use the **Add Job Request** dialog to add the job request for the second job as follows:

Enter the **Request Name** for the second job request, then select option **A** to access the **Add Job Request** panel. The **Owner-ID,** once again, defaults to the TSO User-ID.

```
05/01/26 16:12 -------  PJS - JOB REQUEST MENU  ------------ Job request added
OPTION ===> A

Select one of the following functions:

  L  LIST     - List all Job Requests for Owner
  A  ADD      - Add a new Job Request
  D  DELETE   - Delete a Job Request
  M  MODIFY   - Modify a Job Request
  S  DISPLAY  - Display a Job Request


Request-ID:
  Owner-ID     ===> TCHTST
  Request Name ===> EX4JOB2   (Required for options D, M, and S)


Enter Request-ID (if required) and make a selection, then press ENTER.
```

Press **PF3**

From the **Add Job Request** panel, enter the **Request Description**, then enter the **JCL** command to access the **Specify Job Request JCL Source** panel.

```
05/01/26 15:06 -------  PJS - ADD JOB REQUEST  --------------------------------
COMMAND ===> JCL_____________________________________________________________

Request-ID   ===> TCHTST.EX4JOB2
Description  ===> Example 4 - Job 2 - Using Events_________________

Status       ===> WAIT                  Enabled      ===> YES  (Yes or No)

JCL Data Set ===>

Next Run D/T ===>
Window       ===>

Start D/T    ===>                       End D/T      ===>

Frequency    ===> ONCE

Events       ===> NONE

Commands ==> JCL - JCL Source, FREQ - Frequency, EVENT - Events,
             OPT - Submit Options, VAR - Variables,
             END - Complete Add, CANCEL - Cancel Add
```

Press **ENTER**

On the **Specify Job Request JCL Source** panel, enter the **JCL Data Set Name** and the **Member Name** in which the JCL has been placed. The **JCL Save Option** defaults to **NO**.

```
05/01/26 15:28 -------  PJS - SPECIFY JCL SOURCE  ----------------------------
COMMAND ===> ____________________________________________________________________

Request-ID   ===> TCHTST.EX4JOB2       Status       ===> WAIT
Description  ===> Example 4 - Job 2 - Using Events


JCL Data Set Name ===> TCHTST.MISC.JCL_____________________________
    Member Name   ===> JOB2____

JCL Save Option   ===> NO_____       (Yes, No, or Refresh)





Commands ==> BROWSE - Browse JCL, BRDATA - Browse JCL Data Set,
             FREQ - Frequency, EVENT - Events, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
```

Press **ENTER**

After the information on the **Specify Job Request JCL Source** panel has been successfully entered, enter the **FREQ** command to access the **Specify Job Request Frequency** panel.

```
05/01/26 15:28 -------  PJS - SPECIFY JCL SOURCE  ----------------------------
COMMAND ===> FREQ_____________________________________________________________

Request-ID   ===> TCHTST.EX4JOB2       Status       ===> WAIT
Description  ===> Example 4 - Job 2 - Using Events


JCL Data Set Name ===> TCHTST.MISC.JCL_____________________________
    Member Name   ===> JOB2____

JCL Save Option   ===> NO_____      (Yes, No, or Refresh)





Commands ==> BROWSE - Browse JCL, BRDATA - Browse JCL Data Set,
             FREQ - Frequency, EVENT - Events, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
```

Press **ENTER**

On the **Specify Job Request Frequency** panel, enter the **Start Date and Time**. In this case, the **Start Date** is entered as a **relative date**, to specify 'today'. The Start Time is entered as an **absolute time** to specify the specific time at which the job is to be run. The **Frequency** will default to **Once**.

```
05/01/26 15:22 -------  PJS - SPECIFY JOB FREQUENCY  -------------------------
COMMAND ===> ______________________________________________________________________

Request-ID   ===> TCHTST.EX4JOB2       Status       ===> WAIT
Description  ===> Example 4 - Job 2 - Using Events

Next Run D/T ===>                      Recalculate  ===> ___  (Yes or No)

Start D/T    ===> *________ 06:00 pm   End D/T      ===> _________ ________

Frequency (choose one of the options below)
   Periodic     ===> ___ (num) _______ (units - Yr, Mo, Wk, Day, Hr, or Min)
      or
   Day of Week  ===> _ Sun  _ Mon  _ Tue  _ Wed  _ Thu  _ Fri  _ Sat
      or
   End of Month ===> Last Day - __   (days before last day of each month)
      or
   Calendar(s)  ===> _________________ _________________ _________________
      or
   Once (if none of the above is specified)

Commands ==> JCL - JCL Source, EVENT - Events, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
```

Press **ENTER**

After pressing **ENTER**, the **Next Run Date and Time** is calculated. Also note that the **relative date** we entered for the **Start Date** has been translated to an **absolute date**.

After the information on the **Specify Job Request Frequency** panel has been successfully entered, enter the **EVENT** command to access the **Specify Job Request Events** panel.

```
05/01/26 15:22 -------  PJS - SPECIFY JOB FREQUENCY  --------------------------
COMMAND ===> EVENT____________________________________________________________

Request-ID   ===> TCHTST.EX4JOB2       Status       ===> WAIT
Description  ===> Example 4 - Job 2 - Using Events

Next Run D/T ===> 01/26/2005 18:00     Recalculate  ===> ___  (Yes or No)

Start D/T    ===> 01/26/2005 18:00     End D/T      ===> _________ ________

Frequency (choose one of the options below)
   Periodic     ===> ___ (num) _______ (units - Yr, Mo, Wk, Day, Hr, or Min)
      or
   Day of Week  ===> _ Sun  _ Mon  _ Tue  _ Wed  _ Thu  _ Fri  _ Sat
      or
   End of Month ===> Last Day - __   (days before last day of each month)
      or
   Calendar(s)  ===> _________________ _________________ _________________
      or
   Once (if none of the above is specified)

Commands ==> JCL - JCL Source, EVENT - Events, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
```

Press **ENTER**

On the **Specify Job Request Events** panel, enter the **Event-ID** of the event on which this job depends (**JOB1END**).

```
05/01/26 15:22 ------  PJS - SPECIFY JOB REQUEST EVENTS  --- LINE 1    OF 0
COMMAND ===> ____________________________________________________ SCROLL ===> PAGE

Request-ID   ===> TCHTST.EX4JOB2       Status       ===> WAIT
Description  ===> Example 4 - Job 2 - Using Events

S      Event-ID     Prepost  Date/Time Posted
_  job1end_________  ___
_  ________________  ___
_  ________________  ___
_  ________________  ___
_  ________________  ___
_  ________________  ___
_  ________________  ___
_  ________________  ___
_  ________________  ___
_  ________________  ___

Commands ==> JCL - JCL Source, FREQ - Frequency, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
Line Commands ==> D - Delete Event, P - Post Job Event, R - Reset Job Event
```

Press **ENTER**

After pressing **ENTER**, the **Event-ID** is protected. After the information on the **Specify Job Request Events** panel has been successfully entered, return to the **Add Job Request** panel by entering the **END** command (we will use **PF3**).

```
05/01/26 15:22 ------  PJS - SPECIFY JOB REQUEST EVENTS  --- LINE 1    OF 1
COMMAND ===> ____________________________________________________ SCROLL ===> PAGE

Request-ID   ===> TCHTST.EX4JOB2       Status       ===> WAIT
Description  ===> Example 4 - Job 2 - Using Events

S      Event-ID     Prepost  Date/Time Posted
_  TCHTST.JOB1END     YES
_  ________________   ___
_  ________________   ___
_  ________________   ___
_  ________________   ___
_  ________________   ___
_  ________________   ___
_  ________________   ___
_  ________________   ___
_  ________________   ___

Commands ==> JCL - JCL Source, FREQ - Frequency, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
Line Commands ==> D - Delete Event, P - Post Job Event, R - Reset Job Event
```

Press **PF3**

We have now entered all the parameters we need for this job request. The **Add Job Request** panel has been updated to show the parameters we have entered. We now complete the add and return to the **Job Request Menu** panel by entering the **END** command (once again, we use **PF3**).

```
05/01/26 16:08 -------  PJS - ADD JOB REQUEST  --------------------------------
COMMAND ===> ____________________________________________________________________

Request-ID   ===> TCHTST.EX4JOB2
Description  ===> Example 4 - Job 2 - Using Events_________________

Status       ===> WAIT                 Enabled      ===> YES  (Yes or No)

JCL Data Set ===> TCHTST.MISC.JCL(JOB2)

Next Run D/T ===> 01/26/2005 18:00
Window       ===>

Start D/T    ===> 01/26/2005 18:00     End D/T      ===>

Frequency    ===> ONCE

Events       ===> 00 OF 01

Commands ==> JCL - JCL Source, FREQ - Frequency, EVENT - Events,
             OPT - Submit Options, VAR - Variables,
             END - Complete Add, CANCEL - Cancel Add
```

Press **PF3**

The second job request add is now complete. PJS will submit this job after the first job completes, (but not before 6:00 PM today).

```
05/01/26 16:12 -------  PJS - JOB REQUEST MENU  ------------ Job request added
OPTION ===> ____________________________________________________________________

Select one of the following functions:

  L  LIST     - List all Job Requests for Owner
  A  ADD      - Add a new Job Request
  D  DELETE   - Delete a Job Request
  M  MODIFY   - Modify a Job Request
  S  DISPLAY  - Display a Job Request


Request-ID:
  Owner-ID     ===> TCHTST__
  Request Name ===> EX4JOB2_   (Required for options D, M, and S)


Enter Request-ID (if required) and make a selection, then press ENTER.
```

Press **PF3**

Example 5: How to Set Up Several dependent Jobs

Problem

In this example, we will expand the use of events to handle a more complex dependent job scheduling scenario. Suppose we wish to submit six jobs that comprise a nightly batch processing system. Each job has the following requirements:

JOB1 does not depend on any other job, and is to be submitted at 10:00 PM.

JOB2 does not depend on any other job, and is to be submitted at 12:00 AM.

JOB3 depends on JOB1, and is to be submitted as soon as JOB1 finishes.

JOB4 depends on JOB2 and cannot be submitted before 1:00 AM.

JOB5 depends on JOB1 and JOB2, and cannot be submitted before 2:00 AM.

JOB6 depends on JOB3, JOB4, and JOB5, and can be submitted as soon as all three have finished.

Assume that the JCL for the jobs is in a PDS named '**TCHTST.MISC.JCL**'. The first job is in the member named **JOB1**, the second job is in the member named **JOB2**, and so on.

To control the job dependancies, an event will be defined for each job, to indicate when that job completes successfully. The event for each job will be named **JOB*n*END**, where *n* is the number of the job (1 through 6).. The JCL for each job will be modified to post its event as the last step, if the previous steps have completed successfully.

Before defining the job requests to PJS, first place the JCL in the specified data set and members. Modify JOB1 to include the PJSPOST step at the end of the job as follows:

```
//JOBn     JOB
//  existing JCL statements
//            .
//            .
//            .
//PJSPOST EXEC PGM=PJSPOST,PARM='TCHTST.JOBnEND',
//             COND=(0,NE)
```

The **PARM** parameter specifies the name of the event to be posted, where ***n*** is the number of the job (1 through 6). The **COND** parameter is used to post the event only if all previous job steps completed with zero return codes.

TSO Solution

Use the **PJREQADD** command to add the job request for JOB1 as follows:

```
 READY
pjreqadd id(ex5job1)
 REQ ADD
set desc('Example 5 - Job 1 - Using Advanced Events')
 REQ ADD
set jcldsn('tchtst.misc.jcl(job1)')
 REQ ADD
set days(1) strtdate(*) strttime(10:00pm)
 REQ ADD
end
 PJS2101I  Job request TCHTST.EX5JOB1 successfully added
 READY
pjreqlst id(ex5job1)
    Request-ID          Description
 TCHTST.EX5JOB1     Example 5 - Job 1 - Using Advanced Events

                        Next Run        Window        Notify    Notify
           Status      Date    Time   Time  Option    Userid    Level
          WAIT      01/25/2005 22:00                 TCHTST    INFO

                   JCL Data Set Name                                Save
          TCHTST.MISC.JCL(JOB1)                                    NOSAVE

             Last Submit          Status Change      Spool      Spool    Retry
            Date      Time       Date      Time      Member   Rec Count  Count
                                01/25/2005 14:17:40                    0      0

          Userid   Group
          TCHTST   WTST

                                         Start              End
             Frequency               Date    Time      Date    Time
            1 DAYS                 01/25/2005 22:00

 READY
```

Next, use the **PJREQADD** command to add the job request for JOB2 as follows:

```
 READY
pjreqadd id(ex5job2)
 REQ ADD
set desc('Example 5 - Job 2 - Using Advanced Events')
 REQ ADD
set jcldsn('tchtst.misc.jcl(job2)')
 REQ ADD
set days(1) strtdate(*+1) strttime(12:00am)
 REQ ADD
end
 PJS2101I  Job request TCHTST.EX5JOB2 successfully added
 READY
pjreqlst id(ex5job2)
    Request-ID          Description
 TCHTST.EX5JOB2     Example 5 - Job 2 - Using Advanced Events

                         Next Run        Window        Notify    Notify
           Status      Date    Time   Time  Option    Userid    Level
          WAIT      01/26/2005 00:00                 TCHTST    INFO

                    JCL Data Set Name                               Save
          TCHTST.MISC.JCL(JOB2)                                    NOSAVE

             Last Submit          Status Change      Spool      Spool    Retry
            Date      Time       Date      Time      Member   Rec Count  Count
                                01/25/2005 14:20:57                    0      0

          Userid   Group
          TCHTST   WTST

                                         Start              End
              Frequency             Date    Time      Date    Time
            1 DAYS               01/26/2005 00:00

 READY
```

Next, use the **PJREQADD** command to add the job request for JOB3 as follows:

```
 READY
pjreqadd id(ex5job3)
 REQ ADD
set desc('Example 5 - Job 3 - Using Advanced Events')
 REQ ADD
set jcldsn('tchtst.misc.jcl(job3)')
 REQ ADD
set days(1) strtdate(*) strttime(10:00pm)
 REQ ADD
event add job1end
 REQ ADD
end
 PJS2101I  Job request TCHTST.EX5JOB3 successfully added
 READY
pjreqlst id(ex5job3)
    Request-ID          Description
 TCHTST.EX5JOB3     Example 5 - Job 3 - Using Advanced Events

                        Next Run        Window        Notify    Notify
           Status      Date    Time   Time  Option    Userid    Level
          WAIT      01/25/2005 22:00                 TCHTST    INFO

                   JCL Data Set Name                                Save
          TCHTST.MISC.JCL(JOB3)                                    NOSAVE

             Last Submit          Status Change      Spool      Spool    Retry
            Date      Time       Date      Time      Member   Rec Count  Count
                                01/25/2005 14:23:41                    0      0

          Userid   Group
          TCHTST   WTST

                                     Start              End
             Frequency             Date    Time      Date    Time
            1 DAYS               01/25/2005 22:00

                             Pre-        Posted
             Event-ID        Post    Date      Time
          TCHTST.JOB1END     YES

 READY
```

Next, use the **PJREQADD** command to add the job request for JOB4 as follows:

```
 READY
pjreqadd id(ex5job4)
 REQ ADD
set desc('Example 5 - Job 4 - Using Advanced Events')
 REQ ADD
set jcldsn('tchtst.misc.jcl(job4)')
 REQ ADD
set days(1) strtdate(*+1) strttime(1:00am)
 REQ ADD
event add job2end
 REQ ADD
end
 PJS2101I  Job request TCHTST.EX5JOB4 successfully added
 READY
pjreqlst id(ex5job4)
    Request-ID          Description
 TCHTST.EX5JOB4     Example 5 - Job 4 - Using Advanced Events

                        Next Run        Window        Notify    Notify
           Status      Date    Time   Time  Option    Userid    Level
          WAIT      01/26/2005 01:00                 TCHTST    INFO

                   JCL Data Set Name                                Save
          TCHTST.MISC.JCL(JOB4)                                    NOSAVE

             Last Submit          Status Change      Spool      Spool    Retry
            Date      Time       Date      Time      Member   Rec Count  Count
                                01/25/2005 14:26:55                    0      0

          Userid   Group
          TCHTST   WTST

                                          Start              End
             Frequency                 Date    Time      Date    Time
            1 DAYS                  01/26/2005 01:00

                                Pre-        Posted
             Event-ID           Post    Date      Time
          TCHTST.JOB2END        YES

 READY
```

Next, use the **PJREQADD** command to add the job request for JOB5 as follows:

```
 READY
pjreqadd id(ex5job5)
 REQ ADD
set desc('Example 5 - Job 5 - Using Advanced Events')
 REQ ADD
set jcldsn('tchtst.misc.jcl(job5)')
 REQ ADD
set days(1) strtdate(*+1) strttime(2:00am)
 REQ ADD
event add job1end
 REQ ADD
event add job2end
 REQ ADD
end
 PJS2101I  Job request TCHTST.EX5JOB5 successfully added
 READY
pjreqlst id(ex5job5)
    Request-ID          Description
 TCHTST.EX5JOB5     Example 5 - Job 5 - Using Advanced Events

                        Next Run        Window        Notify    Notify
           Status      Date    Time   Time  Option    Userid    Level
          WAIT      01/26/2005 02:00                 TCHTST    INFO

                   JCL Data Set Name                                Save
          TCHTST.MISC.JCL(JOB5)                                    NOSAVE

             Last Submit          Status Change      Spool      Spool    Retry
            Date      Time       Date      Time      Member   Rec Count  Count
                                01/25/2005 14:30:11                    0      0

          Userid   Group
          TCHTST   WTST

                                     Start              End
              Frequency            Date    Time      Date    Time
            1 DAYS               01/26/2005 02:00

                             Pre-        Posted
              Event-ID       Post    Date      Time
          TCHTST.JOB1END     YES
          TCHTST.JOB2END     YES

 READY
```

Next, use the **PJREQADD** command to add the job request for JOB6 as follows:

```
 READY
pjreqadd id(ex5job6)
 REQ ADD
set desc('Example 5 - Job 6 - Using Advanced Events')
 REQ ADD
set jcldsn('tchtst.misc.jcl(job6)')
 REQ ADD
set days(1) strtdate(*) strttime(10:00pm)
 REQ ADD
event add job3end
 REQ ADD
event add job4end
 REQ ADD
event add job5end
 REQ ADD
end
 PJS2101I  Job request TCHTST.EX5JOB6 successfully added
 READY
pjreqlst id(ex5job6)
    Request-ID          Description
 TCHTST.EX5JOB6     Example 5 - Job 6 - Using Advanced Events

                        Next Run        Window        Notify    Notify
           Status      Date    Time   Time  Option    Userid    Level
          WAIT      01/25/2005 22:00                 TCHTST    INFO

                   JCL Data Set Name                                Save
          TCHTST.MISC.JCL(JOB6)                                    NOSAVE

             Last Submit          Status Change      Spool      Spool    Retry
            Date      Time       Date      Time      Member   Rec Count  Count
                                01/25/2005 14:34:00                    0      0

          Userid   Group
          TCHTST   WTST

                                         Start              End
             Frequency              Date    Time      Date    Time
            1 DAYS               01/25/2005 22:00

                              Pre-        Posted
             Event-ID         Post    Date      Time
          TCHTST.JOB3END      YES
          TCHTST.JOB4END      YES
          TCHTST.JOB5END      YES

 READY
```

ISPF Solution

Use the **Add Job Request** dialog to add the job request for JOB1 as follows:

Access the **PJS Main Menu** panel from your **ISPF Primary Option Menu** panel, or by what ever other means your installation uses. Then select option **J** to access the **PJS Job Request Menu** panel.

```
 05/01/26 14:49 -------  PJS - PERSONAL JOB SCHEDULER  ---------------- 3.1.0
 OPTION ===> J_____________________________________________________________

Select one of the following functions:

  J  JOBREQ   - Update Job Requests
  C  CALENDAR - Update Calendars
  E  EVENT    - Update Events
  V  VARIABLE - Update Variables

  T  TUTORIAL - Enter PJS/ISPF Tutorial


Select an option, then press ENTER.


(c) Copyright, Northrop Grumman, 1990, 2005.
For additional copyright information enter ABOUT on the command line.

PJS is distributed under the GNU General Public License.
For license information enter LICENSE on the command line.
```

Press **ENTER**

On the **PJS Job Request Menu** panel, enter the **Request Name**, then select option **A** to access the **Add Job Request** panel. The **Owner-ID** defaults to the TSO User-ID.

```
05/01/26 14:49 -------  PJS - JOB REQUEST MENU  -------------------------------
OPTION ===> A____________________________________________________________________

Select one of the following functions:

  L  LIST     - List all Job Requests for Owner
  A  ADD      - Add a new Job Request
  D  DELETE   - Delete a Job Request
  M  MODIFY   - Modify a Job Request
  S  DISPLAY  - Display a Job Request


Request-ID:
  Owner-ID     ===> TCHTST__
  Request Name ===> EX5JOB1_   (Required for options D, M, and S)


Enter Request-ID (if required) and make a selection, then press ENTER.
```

Press **ENTER**

From the **Add Job Request** panel, enter the **Request Description**, then enter the **JCL** command to access the **Specify Job Request JCL Source** panel.

```
05/01/26 14:49 -------  PJS - ADD JOB REQUEST  --------------------------------
COMMAND ===> JCL_____________________________________________________________

Request-ID   ===> TCHTST.EX5JOB1
Description  ===> Example 5 - Job 1 - Using Advanced Events_________

Status       ===> WAIT                  Enabled      ===> YES  (Yes or No)

JCL Data Set ===>

Next Run D/T ===>
Window       ===>

Start D/T    ===>                       End D/T      ===>

Frequency    ===> ONCE

Events       ===> NONE

Commands ==> JCL - JCL Source, FREQ - Frequency, EVENT - Events,
             OPT - Submit Options, VAR - Variables,
             END - Complete Add, CANCEL - Cancel Add
```

Press **ENTER**

On the **Specify Job Request JCL Source** panel, enter the **JCL Data Set Name** and the **Member Name** in which the JCL has been placed. The **JCL Save Option** defaults to **NO**.

```
05/01/26 14:50 -------  PJS - SPECIFY JCL SOURCE  ----------------------------
COMMAND ===> ____________________________________________________________________

Request-ID   ===> TCHTST.EX5JOB1       Status       ===> WAIT
Description  ===>  Example 5 - Job 1 - Using Advanced Events


JCL Data Set Name ===> TCHTST.MISC.JCL_____________________________
    Member Name   ===> JOB1____

JCL Save Option   ===> NO_____       (Yes, No, or Refresh)





Commands ==> BROWSE - Browse JCL, BRDATA - Browse JCL Data Set,
             FREQ - Frequency, EVENT - Events, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
```

Press **ENTER**

After the information on the **Specify Job Request JCL Source** panel has been successfully entered, enter the **FREQ** command to access the **Specify Job Request Frequency** panel.

```
05/01/26 14:50 -------  PJS - SPECIFY JCL SOURCE  ----------------------------
COMMAND ===> FREQ_____________________________________________________________

Request-ID   ===> TCHTST.EX5JOB1       Status       ===> WAIT
Description  ===> Example 5 - Job 1 - Using Advanced Events


JCL Data Set Name ===> TCHTST.MISC.JCL_____________________________
    Member Name   ===> JOB1____

JCL Save Option   ===> NO_____      (Yes, No, or Refresh)





Commands ==> BROWSE - Browse JCL, BRDATA - Browse JCL Data Set,
             FREQ - Frequency, EVENT - Events, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
```

Press **ENTER**

On the **Specify Job Request Frequency** panel, enter the **Start Date and Time**. In this case, the **Start Date** is entered as a **relative date**, to specify 'today'. The Start Time is entered as an **absolute time** to specify the specific time at which the job is to be run. The **Frequency** is entered as a **Periodic** frequency by entering **1** in the number field, and **DAYS** in the units field.

```
05/01/26 14:50 -------  PJS - SPECIFY JOB FREQUENCY  --------------------------
COMMAND ===> ______________________________________________________________________

Request-ID   ===> TCHTST.EX5JOB1       Status       ===> WAIT
Description  ===> Example 5 - Job 1 - Using Advanced Events

Next Run D/T ===>                      Recalculate  ===> ___  (Yes or No)

Start D/T    ===> *________ 10:00 pm  End D/T      ===> _________ ________

Frequency (choose one of the options below)
   Periodic     ===> 1__ (num) days___ (units - Yr, Mo, Wk, Day, Hr, or Min)
      or
   Day of Week  ===> _ Sun  _ Mon  _ Tue  _ Wed  _ Thu  _ Fri  _ Sat
      or
   End of Month ===> Last Day - __   (days before last day of each month)
      or
   Calendar(s)  ===> _________________ _________________ _________________
      or
   Once (if none of the above is specified)

Commands ==> JCL - JCL Source, EVENT - Events, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
```

Press **ENTER**

After pressing **ENTER**, the **Next Run Date and Time** is calculated. Also note that the **relative date** we entered for the **Start Date** has been translated to an **absolute date**. After the information on the **Specify Job Request JCL Source** panel has been successfully entered, return to the **Add Job Request** panel by entering the **END** command (we will use **PF3**).

```
05/01/26 14:50 -------  PJS - SPECIFY JOB FREQUENCY  --------------------------
COMMAND ===> ____________________________________________________________________

Request-ID   ===> TCHTST.EX5JOB1       Status       ===> WAIT
Description  ===> Example 5 - Job 1 - Using Advanced Events

Next Run D/T ===> 01/26/2005 22:00     Recalculate  ===> ___  (Yes or No)

Start D/T    ===> 01/26/2005 22:00     End D/T      ===> _________ ________

Frequency (choose one of the options below)
   Periodic     ===> 001 (num) DAYS___ (units - Yr, Mo, Wk, Day, Hr, or Min)
      or
   Day of Week  ===> _ Sun  _ Mon  _ Tue  _ Wed  _ Thu  _ Fri  _ Sat
      or
   End of Month ===> Last Day - __   (days before last day of each month)
      or
   Calendar(s)  ===> _________________ _________________ _________________
      or
   Once (if none of the above is specified)

Commands ==> JCL - JCL Source, EVENT - Events, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
```

Press **PF3**

We have now entered all the parameters we need for this job request. The **Add Job Request** panel has been updated to show the parameters we have entered. We now complete the add and return to the **Job Request Menu** panel by entering the **END** command (once again, we use **PF3**).

```
05/01/26 14:50 -------  PJS - ADD JOB REQUEST  --------------------------------
COMMAND ===> ____________________________________________________________________

Request-ID   ===> TCHTST.EX5JOB1
Description  ===> Example 5 - Job 1 - Using Advanced Events_________

Status       ===> WAIT                 Enabled      ===> YES  (Yes or No)

JCL Data Set ===> TCHTST.MISC.JCL(JOB1)

Next Run D/T ===> 01/26/2005 22:00
Window       ===>

Start D/T    ===> 01/26/2005 22:00     End D/T      ===>

Frequency    ===> 001 DAYS

Events       ===> NONE

Commands ==> JCL - JCL Source, FREQ - Frequency, EVENT - Events,
             OPT - Submit Options, VAR - Variables,
             END - Complete Add, CANCEL - Cancel Add
```

Press **PF3**

The job request for JOB1 is now complete. PJS will submit this job every day at 10:00 PM.

Next, use the **Add Job Request** dialog to add the job request for JOB2 as follows:

Enter the **Request Name** for the second job request, then select option **A** to access the **Add Job Request** panel. The **Owner-ID,** once again, defaults to the TSO User-ID.

```
05/01/26 14:50 -------  PJS - JOB REQUEST MENU  ------------ Job request added
OPTION ===> A

Select one of the following functions:

  L  LIST     - List all Job Requests for Owner
  A  ADD      - Add a new Job Request
  D  DELETE   - Delete a Job Request
  M  MODIFY   - Modify a Job Request
  S  DISPLAY  - Display a Job Request


Request-ID:
  Owner-ID     ===> TCHTST
  Request Name ===> EX5JOB2   (Required for options D, M, and S)


Enter Request-ID (if required) and make a selection, then press ENTER.
```

Press **PF3**

From the **Add Job Request** panel, enter the **Request Description**, then enter the **JCL** command to access the **Specify Job Request JCL Source** panel.

```
05/01/26 14:50 -------  PJS - ADD JOB REQUEST  --------------------------------
COMMAND ===> JCL_____________________________________________________________

Request-ID   ===> TCHTST.EX5JOB2
Description  ===> Example 5 - Job 2 - Using Advanced Events_________

Status       ===> WAIT                 Enabled      ===> YES  (Yes or No)

JCL Data Set ===>

Next Run D/T ===>
Window       ===>

Start D/T    ===>                      End D/T      ===>

Frequency    ===> ONCE

Events       ===> NONE

Commands ==> JCL - JCL Source, FREQ - Frequency, EVENT - Events,
             OPT - Submit Options, VAR - Variables,
             END - Complete Add, CANCEL - Cancel Add
```

Press **ENTER**

On the **Specify Job Request JCL Source** panel, enter the **JCL Data Set Name** and the **Member Name** in which the JCL has been placed. The **JCL Save Option** defaults to **NO**.

```
05/01/26 14:51 -------  PJS - SPECIFY JCL SOURCE  ----------------------------
COMMAND ===> ____________________________________________________________________

Request-ID   ===> TCHTST.EX5JOB2       Status       ===> WAIT
Description  ===> Example 5 - Job 2 - Using Advanced Events


JCL Data Set Name ===> TCHTST.MISC.JCL_____________________________
    Member Name   ===> JOB2____

JCL Save Option   ===> NO_____      (Yes, No, or Refresh)





Commands ==> BROWSE - Browse JCL, BRDATA - Browse JCL Data Set,
             FREQ - Frequency, EVENT - Events, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
```

Press **ENTER**

After the information on the **Specify Job Request JCL Source** panel has been successfully entered, enter the **FREQ** command to access the **Specify Job Request Frequency** panel.

```
05/01/26 14:51 -------  PJS - SPECIFY JCL SOURCE  ----------------------------
COMMAND ===> FREQ_____________________________________________________________

Request-ID   ===> TCHTST.EX5JOB2       Status       ===> WAIT
Description  ===> Example 5 - Job 2 - Using Advanced Events


JCL Data Set Name ===> TCHTST.MISC.JCL_____________________________
    Member Name   ===> JOB2____

JCL Save Option   ===> NO_____      (Yes, No, or Refresh)





Commands ==> BROWSE - Browse JCL, BRDATA - Browse JCL Data Set,
             FREQ - Frequency, EVENT - Events, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
```

Press **ENTER**

On the **Specify Job Request Frequency** panel, enter the **Start Date and Time**. In this case, the **Start Date** is entered as a **relative date**, to specify 'today'. The Start Time is entered as an **absolute time** to specify the specific time at which the job is to be run. The **Frequency** is entered as a **Periodic** frequency by entering **1** in the number field, and **DAYS** in the units field.

```
05/01/26 14:51 -------  PJS - SPECIFY JOB FREQUENCY  --------------------------
COMMAND ===> ____________________________________________________________________

Request-ID   ===> TCHTST.EX5JOB2       Status       ===> WAIT
Description  ===> Example 5 - Job 2 - Using Advanced Events

Next Run D/T ===>                      Recalculate  ===> ___  (Yes or No)

Start D/T    ===> *+1_______ 12:00 am  End D/T      ===> __________ ________

Frequency (choose one of the options below)
   Periodic     ===> 1__ (num) days___ (units - Yr, Mo, Wk, Day, Hr, or Min)
      or
   Day of Week  ===> _ Sun  _ Mon  _ Tue  _ Wed  _ Thu  _ Fri  _ Sat
      or
   End of Month ===> Last Day - __   (days before last day of each month)
      or
   Calendar(s)  ===> _________________ _________________ _________________
      or
   Once (if none of the above is specified)

Commands ==> JCL - JCL Source, EVENT - Events, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
```

Press **ENTER**

After pressing **ENTER**, the **Next Run Date and Time** is calculated. Also note that the **relative date** we entered for the **Start Date** has been translated to an **absolute date**.

After the information on the **Specify Job Request Frequency** panel has been successfully entered, return to the **Add Job Request** panel by entering the **END** command (once again, we will use **PF3).**

```
05/01/26 14:52 -------  PJS - SPECIFY JOB FREQUENCY  --------------------------
COMMAND ===> ____________________________________________________________________

Request-ID   ===> TCHTST.EX5JOB2       Status       ===> WAIT
Description  ===> Example 5 - Job 2 - Using Advanced Events

Next Run D/T ===> 01/27/2005 00:00     Recalculate  ===> ___  (Yes or No)

Start D/T    ===> 01/27/2005 00:00     End D/T      ===> _________ ________

Frequency (choose one of the options below)
   Periodic     ===> ___ (num) _______ (units - Yr, Mo, Wk, Day, Hr, or Min)
      or
   Day of Week  ===> _ Sun  _ Mon  _ Tue  _ Wed  _ Thu  _ Fri  _ Sat
      or
   End of Month ===> Last Day - __   (days before last day of each month)
      or
   Calendar(s)  ===> _________________ _________________ _________________
      or
   Once (if none of the above is specified)

Commands ==> JCL - JCL Source, EVENT - Events, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
```

Press **PF3**

We have now entered all the parameters we need for this job request. The **Add Job Request** panel has been updated to show the parameters we have entered. We now complete the add and return to the **Job Request Menu** panel by entering the **END** command (once again, we use **PF3**).

```
05/01/26 14:52 -------  PJS - ADD JOB REQUEST  --------------------------------
COMMAND ===> ____________________________________________________________________

Request-ID   ===> TCHTST.EX5JOB2
Description  ===> Example 5 - Job 2 - Using Advanced Events________

Status       ===> WAIT                 Enabled      ===> YES  (Yes or No)

JCL Data Set ===> TCHTST.MISC.JCL(JOB2)

Next Run D/T ===> 01/27/2005 00:00
Window       ===>

Start D/T    ===> 01/27/2005 00:00     End D/T      ===>

Frequency    ===> ONCE

Events       ===> NONE

Commands ==> JCL - JCL Source, FREQ - Frequency, EVENT - Events,
             OPT - Submit Options, VAR - Variables,
             END - Complete Add, CANCEL - Cancel Add
```

Press **PF3**

The job request add for JOB2 is now complete. PJS will submit this job every day at 12:00 AM.

Next, use the **Add Job Request** dialog to add the job request for JOB3 as follows:

Enter the **Request Name** for the second job request, then select option **A** to access the **Add Job Request** panel. The **Owner-ID,** once again, defaults to the TSO User-ID.

```
05/01/26 14:52 -------  PJS - JOB REQUEST MENU  ------------ Job request added
OPTION ===> A

Select one of the following functions:

  L  LIST     - List all Job Requests for Owner
  A  ADD      - Add a new Job Request
  D  DELETE   - Delete a Job Request
  M  MODIFY   - Modify a Job Request
  S  DISPLAY  - Display a Job Request


Request-ID:
  Owner-ID     ===> TCHTST
  Request Name ===> EX5JOB3   (Required for options D, M, and S)


Enter Request-ID (if required) and make a selection, then press ENTER.
```

Press **PF3**

From the **Add Job Request** panel, enter the **Request Description**, then enter the **JCL** command to access the **Specify Job Request JCL Source** panel.

```
05/01/26 14:53 -------  PJS - ADD JOB REQUEST  --------------------------------
COMMAND ===> JCL_____________________________________________________________

Request-ID   ===> TCHTST.EX5JOB3
Description  ===> Example 5 - Job 3 - Using Advanced Events________

Status       ===> WAIT                  Enabled      ===> YES  (Yes or No)

JCL Data Set ===>

Next Run D/T ===>
Window       ===>

Start D/T    ===>                       End D/T      ===>

Frequency    ===> ONCE

Events       ===> NONE

Commands ==> JCL - JCL Source, FREQ - Frequency, EVENT - Events,
             OPT - Submit Options, VAR - Variables,
             END - Complete Add, CANCEL - Cancel Add
```

Press **ENTER**

On the **Specify Job Request JCL Source** panel, enter the **JCL Data Set Name** and the **Member Name** in which the JCL has been placed. The **JCL Save Option** defaults to **NO**.

```
05/01/26 14:53 -------  PJS - SPECIFY JCL SOURCE  ----------------------------
COMMAND ===> ____________________________________________________________________

Request-ID   ===> TCHTST.EX5JOB3       Status       ===> WAIT
Description  ===> Example 5 - Job 3 - Using Advanced Events


JCL Data Set Name ===> TCHTST.MISC.JCL_____________________________
    Member Name   ===> JOB3____

JCL Save Option   ===> NO_____       (Yes, No, or Refresh)





Commands ==> BROWSE - Browse JCL, BRDATA - Browse JCL Data Set,
             FREQ - Frequency, EVENT - Events, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
```

Press **ENTER**

After the information on the **Specify Job Request JCL Source** panel has been successfully entered, enter the **FREQ** command to access the **Specify Job Request Frequency** panel.

```
05/01/26 14:53 -------  PJS - SPECIFY JCL SOURCE  ----------------------------
COMMAND ===> FREQ_____________________________________________________________

Request-ID   ===> TCHTST.EX5JOB3       Status       ===> WAIT
Description  ===> Example 5 - Job 3 - Using Advanced Events


JCL Data Set Name ===> TCHTST.MISC.JCL_____________________________
    Member Name   ===> JOB3____

JCL Save Option   ===> NO_____      (Yes, No, or Refresh)





Commands ==> BROWSE - Browse JCL, BRDATA - Browse JCL Data Set,
             FREQ - Frequency, EVENT - Events, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
```

Press **ENTER**

On the **Specify Job Request Frequency** panel, enter the **Start Date and Time**. In this case, the **Start Date** is entered as a **relative date**, to specify 'today'. The Start Time is entered as an **absolute time** to specify the specific time at which the job is to be run. The **Frequency** is entered as a **Periodic** frequency by entering **1** in the number field, and **DAYS** in the units field.

```
05/01/26 14:53 -------  PJS - SPECIFY JOB FREQUENCY  --------------------------
COMMAND ===> ____________________________________________________________________

Request-ID   ===> TCHTST.EX5JOB3       Status       ===> WAIT
Description  ===> Example 5 - Job 3 - Using Advanced Events

Next Run D/T ===>                      Recalculate  ===> ___  (Yes or No)

Start D/T    ===> *________ 10:00 pm  End D/T      ===> _________ ________

Frequency (choose one of the options below)
   Periodic     ===> 1__ (num) days___ (units - Yr, Mo, Wk, Day, Hr, or Min)
      or
   Day of Week  ===> _ Sun  _ Mon  _ Tue  _ Wed  _ Thu  _ Fri  _ Sat
      or
   End of Month ===> Last Day - __   (days before last day of each month)
      or
   Calendar(s)  ===> _______________ _______________ _______________
      or
   Once (if none of the above is specified)

Commands ==> JCL - JCL Source, EVENT - Events, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
```

Press **ENTER**

After pressing **ENTER**, the **Next Run Date and Time** is calculated. Also note that the **relative date** we entered for the **Start Date** has been translated to an **absolute date**.

After the information on the **Specify Job Request Frequency** panel has been successfully entered, enter the **EVENT** command to access the **Specify Job Request Events** panel.

```
05/01/26 14:53 -------  PJS - SPECIFY JOB FREQUENCY  --------------------------
COMMAND ===> EVENT____________________________________________________________

Request-ID   ===> TCHTST.EX5JOB3       Status       ===> WAIT
Description  ===> Example 5 - Job 3 - Using Advanced Events

Next Run D/T ===> 01/26/2005 22:00     Recalculate  ===> ___  (Yes or No)

Start D/T    ===> 01/26/2005 22:00     End D/T      ===> _________ ________

Frequency (choose one of the options below)
   Periodic     ===> 001 (num) DAYS___ (units - Yr, Mo, Wk, Day, Hr, or Min)
      or
   Day of Week  ===> _ Sun  _ Mon  _ Tue  _ Wed  _ Thu  _ Fri  _ Sat
      or
   End of Month ===> Last Day - __   (days before last day of each month)
      or
   Calendar(s)  ===> _________________ _________________ _________________
      or
   Once (if none of the above is specified)

Commands ==> JCL - JCL Source, EVENT - Events, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
```

Press **ENTER**

On the **Specify Job Request Events** panel, enter the **Event-ID** of the event on which this job depends (**JOB1END**).

```
05/01/26 14:54 ------  PJS - SPECIFY JOB REQUEST EVENTS  --- LINE 1    OF 0
COMMAND ===> ____________________________________________________ SCROLL ===> PAGE

Request-ID   ===> TCHTST.EX5JOB3       Status       ===> WAIT
Description  ===> Example 5 - Job 3 - Using Advanced Events

S      Event-ID     Prepost  Date/Time Posted
_  job1end_________  ___
_  ________________  ___
_  ________________  ___
_  ________________  ___
_  ________________  ___
_  ________________  ___
_  ________________  ___
_  ________________  ___
_  ________________  ___
_  ________________  ___

Commands ==> JCL - JCL Source, FREQ - Frequency, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
Line Commands ==> D - Delete Event, P - Post Job Event, R - Reset Job Event
```

Press **ENTER**

After pressing **ENTER**, the **Event-ID** is protected. After the information on the **Specify Job Request Events** panel has been successfully entered, return to the **Add Job Request** panel by entering the **END** command (we will use **PF3**).

```
05/01/26 14:54 ------  PJS - SPECIFY JOB REQUEST EVENTS  --- LINE 1    OF 1
COMMAND ===> ____________________________________________________ SCROLL ===> PAGE

Request-ID   ===> TCHTST.EX5JOB3       Status       ===> WAIT
Description  ===> Example 5 - Job 3 - Using Advanced Events

S      Event-ID     Prepost  Date/Time Posted
_  TCHTST.JOB1END     YES
_  ________________   ___
_  ________________   ___
_  ________________   ___
_  ________________   ___
_  ________________   ___
_  ________________   ___
_  ________________   ___
_  ________________   ___
_  ________________   ___

Commands ==> JCL - JCL Source, FREQ - Frequency, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
Line Commands ==> D - Delete Event, P - Post Job Event, R - Reset Job Event
```

Press **PF3**

We have now entered all the parameters we need for this job request. The **Add Job Request** panel has been updated to show the parameters we have entered. We now complete the add and return to the **Job Request Menu** panel by entering the **END** command (once again, we use **PF3**).

```
05/01/26 14:54 -------  PJS - ADD JOB REQUEST  --------------------------------
COMMAND ===> ____________________________________________________________________

Request-ID   ===> TCHTST.EX5JOB3
Description  ===> Example 5 - Job 3 - Using Advanced Events________

Status       ===> WAIT                 Enabled      ===> YES  (Yes or No)

JCL Data Set ===> TCHTST.MISC.JCL(JOB3)

Next Run D/T ===> 01/26/2005 22:00
Window       ===>

Start D/T    ===> 01/26/2005 22:00     End D/T      ===>

Frequency    ===> ONCE

Events       ===> 00 OF 01

Commands ==> JCL - JCL Source, FREQ - Frequency, EVENT - Events,
             OPT - Submit Options, VAR - Variables,
             END - Complete Add, CANCEL - Cancel Add
```

Press **PF3 twice**

The job request add for JOB3 is now complete. PJS will submit this job every day at 12:00 AM.

Next, use the **Add Job Request** dialog to add the job request for JOB4 as follows:

Enter the **Request Name** for the second job request, then select option **A** to access the **Add Job Request** panel. The **Owner-ID,** once again, defaults to the TSO User-ID.

```
05/01/26 14:55 -------  PJS - JOB REQUEST MENU  ------------ Job request added
OPTION ===> A

Select one of the following functions:

  L  LIST     - List all Job Requests for Owner
  A  ADD      - Add a new Job Request
  D  DELETE   - Delete a Job Request
  M  MODIFY   - Modify a Job Request
  S  DISPLAY  - Display a Job Request


Request-ID:
  Owner-ID     ===> TCHTST
  Request Name ===> EX5JOB4   (Required for options D, M, and S)


Enter Request-ID (if required) and make a selection, then press ENTER.
```

Press **PF3**

From the **Add Job Request** panel, enter the **Request Description**, then enter the **JCL** command to access the **Specify Job Request JCL Source** panel.

```
05/01/26 14:55 -------  PJS - ADD JOB REQUEST  --------------------------------
COMMAND ===> JCL_____________________________________________________________

Request-ID   ===> TCHTST.EX5JOB4
Description  ===> Example 5 - Job 4 - Using Advanced Events_________

Status       ===> WAIT                 Enabled      ===> YES  (Yes or No)

JCL Data Set ===>

Next Run D/T ===>
Window       ===>

Start D/T    ===>                      End D/T      ===>

Frequency    ===> ONCE

Events       ===> NONE

Commands ==> JCL - JCL Source, FREQ - Frequency, EVENT - Events,
             OPT - Submit Options, VAR - Variables,
             END - Complete Add, CANCEL - Cancel Add
```

Press **ENTER**

On the **Specify Job Request JCL Source** panel, enter the **JCL Data Set Name** and the **Member Name** in which the JCL has been placed. The **JCL Save Option** defaults to **NO**.

```
05/01/26 14:55 -------  PJS - SPECIFY JCL SOURCE  ----------------------------
COMMAND ===> ____________________________________________________________________

Request-ID   ===> TCHTST.EX5JOB4       Status       ===> WAIT
Description  ===> Example 5 - Job 4 - Using Advanced Events


JCL Data Set Name ===> TCHTST.MISC.JCL_____________________________
    Member Name   ===> JOB4____

JCL Save Option   ===> NO_____      (Yes, No, or Refresh)





Commands ==> BROWSE - Browse JCL, BRDATA - Browse JCL Data Set,
             FREQ - Frequency, EVENT - Events, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
```

Press **ENTER**

After the information on the **Specify Job Request JCL Source** panel has been successfully entered, enter the **FREQ** command to access the **Specify Job Request Frequency** panel.

```
05/01/26 14:55 -------  PJS - SPECIFY JCL SOURCE  ----------------------------
COMMAND ===> FREQ_____________________________________________________________

Request-ID   ===> TCHTST.EX5JOB4       Status       ===> WAIT
Description  ===> Example 5 - Job 4 - Using Advanced Events


JCL Data Set Name ===> TCHTST.MISC.JCL_____________________________
    Member Name   ===> JOB4____

JCL Save Option   ===> NO_____      (Yes, No, or Refresh)





Commands ==> BROWSE - Browse JCL, BRDATA - Browse JCL Data Set,
             FREQ - Frequency, EVENT - Events, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
```

Press **ENTER**

On the **Specify Job Request Frequency** panel, enter the **Start Date and Time**. In this case, the **Start Date** is entered as a **relative date**, to specify 'today'. The Start Time is entered as an **absolute time** to specify the specific time at which the job is to be run. The **Frequency** is entered as a **Periodic** frequency by entering **1** in the number field, and **DAYS** in the units field.

```
05/01/26 14:55 -------  PJS - SPECIFY JOB FREQUENCY  --------------------------
COMMAND ===> ____________________________________________________________________

Request-ID   ===> TCHTST.EX5JOB4       Status       ===> WAIT
Description  ===> Example 5 - Job 4 - Using Advanced Events

Next Run D/T ===>                      Recalculate  ===> ___  (Yes or No)

Start D/T    ===> *+1_______ 1:00 am_  End D/T      ===> __________ ________

Frequency (choose one of the options below)
   Periodic     ===> 1__ (num) days___ (units - Yr, Mo, Wk, Day, Hr, or Min)
      or
   Day of Week  ===> _ Sun  _ Mon  _ Tue  _ Wed  _ Thu  _ Fri  _ Sat
      or
   End of Month ===> Last Day - __   (days before last day of each month)
      or
   Calendar(s)  ===> _________________ _________________ _________________
      or
   Once (if none of the above is specified)

Commands ==> JCL - JCL Source, EVENT - Events, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
```

Press **ENTER**

After pressing **ENTER**, the **Next Run Date and Time** is calculated. Also note that the **relative date** we entered for the **Start Date** has been translated to an **absolute date**.

After the information on the **Specify Job Request Frequency** panel has been successfully entered, enter the **EVENT** command to access the **Specify Job Request Events** panel.

```
05/01/26 14:55 -------  PJS - SPECIFY JOB FREQUENCY  --------------------------
COMMAND ===> EVENT____________________________________________________________

Request-ID   ===> TCHTST.EX5JOB4       Status       ===> WAIT
Description  ===> Example 5 - Job 4 - Using Advanced Events

Next Run D/T ===> 01/27/2005 01:00     Recalculate  ===> ___  (Yes or No)

Start D/T    ===> 01/27/2005 01:00     End D/T      ===> _________ ________

Frequency (choose one of the options below)
   Periodic     ===> 001 (num) DAYS___ (units - Yr, Mo, Wk, Day, Hr, or Min)
      or
   Day of Week  ===> _ Sun  _ Mon  _ Tue  _ Wed  _ Thu  _ Fri  _ Sat
      or
   End of Month ===> Last Day - __   (days before last day of each month)
      or
   Calendar(s)  ===> _________________ _________________ _________________
      or
   Once (if none of the above is specified)

Commands ==> JCL - JCL Source, EVENT - Events, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
```

Press **ENTER**

On the **Specify Job Request Events** panel, enter the **Event-ID** of the event on which this job depends (**JOB2END**).

```
05/01/26 14:56 ------  PJS - SPECIFY JOB REQUEST EVENTS  --- LINE 1    OF 0
COMMAND ===> ____________________________________________________ SCROLL ===> PAGE

Request-ID   ===> TCHTST.EX5JOB4       Status       ===> WAIT
Description  ===> Example 5 - Job 4 - Using Advanced Events

S      Event-ID     Prepost  Date/Time Posted
_  job2end_________  ___
_  ________________  ___
_  ________________  ___
_  ________________  ___
_  ________________  ___
_  ________________  ___
_  ________________  ___
_  ________________  ___
_  ________________  ___
_  ________________  ___

Commands ==> JCL - JCL Source, FREQ - Frequency, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
Line Commands ==> D - Delete Event, P - Post Job Event, R - Reset Job Event
```

Press **ENTER**

After pressing **ENTER**, the **Event-ID** is protected. After the information on the **Specify Job Request Events** panel has been successfully entered, return to the **Add Job Request** panel by entering the **END** command (we will use **PF3**).

```
05/01/26 14:56 ------  PJS - SPECIFY JOB REQUEST EVENTS  --- LINE 1    OF 1
COMMAND ===> ____________________________________________________ SCROLL ===> PAGE

Request-ID   ===> TCHTST.EX5JOB4       Status       ===> WAIT
Description  ===> Example 5 - Job 4 - Using Advanced Events

S      Event-ID     Prepost  Date/Time Posted
_   TCHTST.JOB2END     YES
_  ________________  ___
_  ________________  ___
_  ________________  ___
_  ________________  ___
_  ________________  ___
_  ________________  ___
_  ________________  ___
_  ________________  ___
_  ________________  ___

Commands ==> JCL - JCL Source, FREQ - Frequency, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
Line Commands ==> D - Delete Event, P - Post Job Event, R - Reset Job Event
```

Press **PF3**

We have now entered all the parameters we need for this job request. The **Add Job Request** panel has been updated to show the parameters we have entered. We now complete the add and return to the **Job Request Menu** panel by entering the **END** command (once again, we use **PF3**).

```
05/01/26 14:56 -------  PJS - ADD JOB REQUEST  --------------------------------
COMMAND ===> ____________________________________________________________________

Request-ID   ===> TCHTST.EX5JOB4
Description  ===> Example 5 - Job 4 - Using Advanced Events________

Status       ===> WAIT                 Enabled      ===> YES  (Yes or No)

JCL Data Set ===> TCHTST.MISC.JCL(JOB4)

Next Run D/T ===> 01/27/2005 01:00
Window       ===>

Start D/T    ===> 01/27/2005 01:00     End D/T      ===>

Frequency    ===> ONCE

Events       ===> 00 OF 01

Commands ==> JCL - JCL Source, FREQ - Frequency, EVENT - Events,
             OPT - Submit Options, VAR - Variables,
             END - Complete Add, CANCEL - Cancel Add
```

Press **PF3**

The job request add for JOB4 is now complete. PJS will submit this job every day at 1:00 AM.

Next, use the **Add Job Request** dialog to add the job request for JOB5 as follows:

Enter the **Request Name** for the second job request, then select option **A** to access the **Add Job Request** panel. The **Owner-ID,** once again, defaults to the TSO User-ID.

```
05/01/26 14:56 -------  PJS - JOB REQUEST MENU  ------------ Job request added
OPTION ===> A

Select one of the following functions:

  L  LIST     - List all Job Requests for Owner
  A  ADD      - Add a new Job Request
  D  DELETE   - Delete a Job Request
  M  MODIFY   - Modify a Job Request
  S  DISPLAY  - Display a Job Request


Request-ID:
  Owner-ID     ===> TCHTST
  Request Name ===> EX5JOB5   (Required for options D, M, and S)


Enter Request-ID (if required) and make a selection, then press ENTER.
```

Press **PF3**

From the **Add Job Request** panel, enter the **Request Description**, then enter the **JCL** command to access the **Specify Job Request JCL Source** panel.

```
05/01/26 14:57 -------  PJS - ADD JOB REQUEST  --------------------------------
COMMAND ===> JCL_____________________________________________________________

Request-ID   ===> TCHTST.EX5JOB5
Description  ===> Example 5 - Job 5 - Using Advanced Events_________

Status       ===> WAIT                  Enabled      ===> YES  (Yes or No)

JCL Data Set ===>

Next Run D/T ===>
Window       ===>

Start D/T    ===>                       End D/T      ===>

Frequency    ===> ONCE

Events       ===> NONE

Commands ==> JCL - JCL Source, FREQ - Frequency, EVENT - Events,
             OPT - Submit Options, VAR - Variables,
             END - Complete Add, CANCEL - Cancel Add
```

Press **ENTER**

On the **Specify Job Request JCL Source** panel, enter the **JCL Data Set Name** and the **Member Name** in which the JCL has been placed. The **JCL Save Option** defaults to **NO**.

```
05/01/26 14:57 -------  PJS - SPECIFY JCL SOURCE  ----------------------------
COMMAND ===> ____________________________________________________________________

Request-ID   ===> TCHTST.EX5JOB5       Status       ===> WAIT
Description  ===> Example 5 - Job 5 - Using Advanced Events


JCL Data Set Name ===> TCHTST.MISC.JCL_____________________________
    Member Name   ===> JOB5____

JCL Save Option   ===> NO_____       (Yes, No, or Refresh)





Commands ==> BROWSE - Browse JCL, BRDATA - Browse JCL Data Set,
             FREQ - Frequency, EVENT - Events, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
```

Press **ENTER**

After the information on the **Specify Job Request JCL Source** panel has been successfully entered, enter the **FREQ** command to access the **Specify Job Request Frequency** panel.

```
05/01/26 14:57 -------  PJS - SPECIFY JCL SOURCE  ----------------------------
COMMAND ===> FREQ_____________________________________________________________

Request-ID   ===> TCHTST.EX5JOB5       Status       ===> WAIT
Description  ===> Example 5 - Job 5 - Using Advanced Events


JCL Data Set Name ===> TCHTST.MISC.JCL_____________________________
    Member Name   ===> JOB5____

JCL Save Option   ===> NO_____      (Yes, No, or Refresh)





Commands ==> BROWSE - Browse JCL, BRDATA - Browse JCL Data Set,
             FREQ - Frequency, EVENT - Events, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
```

Press **ENTER**

On the **Specify Job Request Frequency** panel, enter the **Start Date and Time**. In this case, the **Start Date** is entered as a **relative date**, to specify 'today'. The Start Time is entered as an **absolute time** to specify the specific time at which the job is to be run. The **Frequency** is entered as a **Periodic** frequency by entering **1** in the number field, and **DAYS** in the units field.

```
05/01/26 14:57 -------  PJS - SPECIFY JOB FREQUENCY  --------------------------
COMMAND ===> ____________________________________________________________________

Request-ID   ===> TCHTST.EX5JOB5       Status       ===> WAIT
Description  ===> Example 5 - Job 5 - Using Advanced Events

Next Run D/T ===>                      Recalculate  ===> ___  (Yes or No)

Start D/T    ===> *+1_______ 2:00 am_  End D/T      ===> __________ ________

Frequency (choose one of the options below)
   Periodic     ===> 1__ (num) days___ (units - Yr, Mo, Wk, Day, Hr, or Min)
      or
   Day of Week  ===> _ Sun  _ Mon  _ Tue  _ Wed  _ Thu  _ Fri  _ Sat
      or
   End of Month ===> Last Day - __   (days before last day of each month)
      or
   Calendar(s)  ===> _________________ _________________ _________________
      or
   Once (if none of the above is specified)

Commands ==> JCL - JCL Source, EVENT - Events, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
```

Press **ENTER**

After pressing **ENTER**, the **Next Run Date and Time** is calculated. Also note that the **relative date** we entered for the **Start Date** has been translated to an **absolute date**.

After the information on the **Specify Job Request Frequency** panel has been successfully entered, enter the **EVENT** command to access the **Specify Job Request Events** panel.

```
05/01/26 14:57 -------  PJS - SPECIFY JOB FREQUENCY  --------------------------
COMMAND ===> EVENT____________________________________________________________

Request-ID   ===> TCHTST.EX5JOB5       Status       ===> WAIT
Description  ===> Example 5 - Job 5 - Using Advanced Events

Next Run D/T ===> 01/27/2005 02:00     Recalculate  ===> ___  (Yes or No)

Start D/T    ===> 01/27/2005 02:00     End D/T      ===> _________ ________

Frequency (choose one of the options below)
   Periodic     ===> 001 (num) DAYS___ (units - Yr, Mo, Wk, Day, Hr, or Min)
      or
   Day of Week  ===> _ Sun  _ Mon  _ Tue  _ Wed  _ Thu  _ Fri  _ Sat
      or
   End of Month ===> Last Day - __   (days before last day of each month)
      or
   Calendar(s)  ===> _________________ _________________ _________________
      or
   Once (if none of the above is specified)

Commands ==> JCL - JCL Source, EVENT - Events, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
```

Press **ENTER**

On the **Specify Job Request Events** panel, enter the **Event-ID** of the events on which this job depends (**JOB1END** and **JOB2END**).

```
05/01/26 14:57 ------  PJS - SPECIFY JOB REQUEST EVENTS  --- LINE 1    OF 0
COMMAND ===> ____________________________________________________ SCROLL ===> PAGE

Request-ID   ===> TCHTST.EX5JOB5       Status       ===> WAIT
Description  ===> Example 5 - Job 5 - Using Advanced Events

S      Event-ID     Prepost  Date/Time Posted
_  job1end_________  ___
_  job2end_________  ___
_  ________________  ___
_  ________________  ___
_  ________________  ___
_  ________________  ___
_  ________________  ___
_  ________________  ___
_  ________________  ___
_  ________________  ___

Commands ==> JCL - JCL Source, FREQ - Frequency, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
Line Commands ==> D - Delete Event, P - Post Job Event, R - Reset Job Event
```

Press **ENTER**

After pressing **ENTER**, the **Event-ID** is protected. After the information on the **Specify Job Request Events** panel has been successfully entered, return to the **Add Job Request** panel by entering the **END** command (we will use **PF3**).

```
05/01/26 14:58 ------  PJS - SPECIFY JOB REQUEST EVENTS  --- LINE 1    OF 2
COMMAND ===> ____________________________________________________ SCROLL ===> PAGE

Request-ID   ===> TCHTST.EX5JOB5       Status       ===> WAIT
Description  ===> Example 5 - Job 5 - Using Advanced Events

S      Event-ID     Prepost  Date/Time Posted
_  TCHTST.JOB1END     YES
_  TCHTST.JOB2END     YES
_  ________________   ___
_  ________________   ___
_  ________________   ___
_  ________________   ___
_  ________________   ___
_  ________________   ___
_  ________________   ___
_  ________________   ___

Commands ==> JCL - JCL Source, FREQ - Frequency, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
Line Commands ==> D - Delete Event, P - Post Job Event, R - Reset Job Event
```

Press **PF3**

We have now entered all the parameters we need for this job request. The **Add Job Request** panel has been updated to show the parameters we have entered. We now complete the add and return to the **Job Request Menu** panel by entering the **END** command (once again, we use **PF3**).

```
05/01/26 14:58 -------  PJS - ADD JOB REQUEST  --------------------------------
COMMAND ===> ____________________________________________________________________

Request-ID   ===> TCHTST.EX5JOB5
Description  ===> Example 5 - Job 5 - Using Advanced Events________

Status       ===> WAIT                 Enabled      ===> YES  (Yes or No)

JCL Data Set ===> TCHTST.MISC.JCL(JOB5)

Next Run D/T ===> 01/27/2005 02:00
Window       ===>

Start D/T    ===> 01/27/2005 02:00     End D/T      ===>

Frequency    ===> ONCE

Events       ===> 00 OF 02

Commands ==> JCL - JCL Source, FREQ - Frequency, EVENT - Events,
             OPT - Submit Options, VAR - Variables,
             END - Complete Add, CANCEL - Cancel Add
```

Press **PF3 twice**

The job request add for JOB5 is now complete. PJS will submit this job every day at 2:00 AM.

Next, use the **Add Job Request** dialog to add the job request for JOB6 as follows:

Enter the **Request Name** for the second job request, then select option **A** to access the **Add Job Request** panel. The **Owner-ID,** once again, defaults to the TSO User-ID.

```
05/01/26 14:58 -------  PJS - JOB REQUEST MENU  ------------ Job request added
OPTION ===> A

Select one of the following functions:

  L  LIST     - List all Job Requests for Owner
  A  ADD      - Add a new Job Request
  D  DELETE   - Delete a Job Request
  M  MODIFY   - Modify a Job Request
  S  DISPLAY  - Display a Job Request


Request-ID:
  Owner-ID     ===> TCHTST
  Request Name ===> EX5JOB6   (Required for options D, M, and S)


Enter Request-ID (if required) and make a selection, then press ENTER.
```

Press **ENTER**

From the **Add Job Request** panel, enter the **Request Description**, then enter the **JCL** command to access the **Specify Job Request JCL Source** panel.

```
05/01/26 14:58 -------  PJS - ADD JOB REQUEST  --------------------------------
COMMAND ===> JCL_____________________________________________________________

Request-ID   ===> TCHTST.EX5JOB6
Description  ===> Example 5 - Job 6 - Using Advanced Events_________

Status       ===> WAIT                 Enabled      ===> YES  (Yes or No)

JCL Data Set ===>

Next Run D/T ===>
Window       ===>

Start D/T    ===>                      End D/T      ===>

Frequency    ===> ONCE

Events       ===> NONE

Commands ==> JCL - JCL Source, FREQ - Frequency, EVENT - Events,
             OPT - Submit Options, VAR - Variables,
             END - Complete Add, CANCEL - Cancel Add
```

Press **ENTER**

On the **Specify Job Request JCL Source** panel, enter the **JCL Data Set Name** and the **Member Name** in which the JCL has been placed. The **JCL Save Option** defaults to **NO**.

```
05/01/26 14:58 -------  PJS - SPECIFY JCL SOURCE  ----------------------------
COMMAND ===> ____________________________________________________________________

Request-ID   ===> TCHTST.EX5JOB6       Status       ===> WAIT
Description  ===> Example 5 - Job 6 - Using Advanced Events


JCL Data Set Name ===> TCHTST.MISC.JCL_____________________________
    Member Name   ===> JOB6____

JCL Save Option   ===> NO_____      (Yes, No, or Refresh)





Commands ==> BROWSE - Browse JCL, BRDATA - Browse JCL Data Set,
             FREQ - Frequency, EVENT - Events, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
```

Press **ENTER**

After the information on the **Specify Job Request JCL Source** panel has been successfully entered, enter the **FREQ** command to access the **Specify Job Request Frequency** panel.

```
05/01/26 14:58 -------  PJS - SPECIFY JCL SOURCE  ----------------------------
COMMAND ===> FREQ_____________________________________________________________

Request-ID   ===> TCHTST.EX5JOB6       Status       ===> WAIT
Description  ===> Example 5 - Job 6 - Using Advanced Events


JCL Data Set Name ===> TCHTST.MISC.JCL_____________________________
    Member Name   ===> JOB6____

JCL Save Option   ===> NO_____      (Yes, No, or Refresh)





Commands ==> BROWSE - Browse JCL, BRDATA - Browse JCL Data Set,
             FREQ - Frequency, EVENT - Events, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
```

Press **ENTER**

On the **Specify Job Request Frequency** panel, enter the **Start Date and Time**. In this case, the **Start Date** is entered as a **relative date**, to specify 'today'. The Start Time is entered as an **absolute time** to specify the specific time at which the job is to be run. The **Frequency** is entered as a **Periodic** frequency by entering **1** in the number field, and **DAYS** in the units field.

```
05/01/26 14:58 -------  PJS - SPECIFY JOB FREQUENCY  --------------------------
COMMAND ===> ____________________________________________________________________

Request-ID   ===> TCHTST.EX5JOB6       Status       ===> WAIT
Description  ===> Example 5 - Job 6 - Using Advanced Events

Next Run D/T ===>                      Recalculate  ===> ___  (Yes or No)

Start D/T    ===> *________ 10:00 pm  End D/T      ===> _________ ________

Frequency (choose one of the options below)
   Periodic     ===> 1__ (num) days___ (units - Yr, Mo, Wk, Day, Hr, or Min)
      or
   Day of Week  ===> _ Sun  _ Mon  _ Tue  _ Wed  _ Thu  _ Fri  _ Sat
      or
   End of Month ===> Last Day - __   (days before last day of each month)
      or
   Calendar(s)  ===> _________________ _________________ _________________
      or
   Once (if none of the above is specified)

Commands ==> JCL - JCL Source, EVENT - Events, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
```

Press **ENTER**

After pressing **ENTER**, the **Next Run Date and Time** is calculated. Also note that the **relative date** we entered for the **Start Date** has been translated to an **absolute date**.

After the information on the **Specify Job Request Frequency** panel has been successfully entered, enter the **EVENT** command to access the **Specify Job Request Events** panel.

```
05/01/26 14:59 -------  PJS - SPECIFY JOB FREQUENCY  --------------------------
COMMAND ===> EVENT_____________________________________________________________

Request-ID   ===> TCHTST.EX5JOB6       Status       ===> WAIT
Description  ===> Example 5 - Job 6 - Using Advanced Events

Next Run D/T ===> 01/26/2005 22:00     Recalculate  ===> ___  (Yes or No)

Start D/T    ===> 01/26/2005 22:00     End D/T      ===> _________ ________

Frequency (choose one of the options below)
   Periodic     ===> 001 (num) DAYS___ (units - Yr, Mo, Wk, Day, Hr, or Min)
      or
   Day of Week  ===> _ Sun  _ Mon  _ Tue  _ Wed  _ Thu  _ Fri  _ Sat
      or
   End of Month ===> Last Day - __   (days before last day of each month)
      or
   Calendar(s)  ===> _________________ _________________ _________________
      or
   Once (if none of the above is specified)

Commands ==> JCL - JCL Source, EVENT - Events, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
```

Press **ENTER**

On the **Specify Job Request Events** panel, enter the **Event-ID** of the events on which this job depends (**JOB3END**, **JOB4END**, and **JOB5END**).

```
05/01/26 14:59 ------  PJS - SPECIFY JOB REQUEST EVENTS  --- LINE 1    OF 0
COMMAND ===> ____________________________________________________ SCROLL ===> PAGE

Request-ID   ===> TCHTST.EX5JOB6       Status       ===> WAIT
Description  ===> Example 5 - Job 6 - Using Advanced Events

S      Event-ID     Prepost  Date/Time Posted
_  job3end_________  ___
_  job4end_________  ___
_  job5end_________  ___
_  ________________  ___
_  ________________  ___
_  ________________  ___
_  ________________  ___
_  ________________  ___
_  ________________  ___
_  ________________  ___

Commands ==> JCL - JCL Source, FREQ - Frequency, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
Line Commands ==> D - Delete Event, P - Post Job Event, R - Reset Job Event
```

Press **ENTER**

After pressing **ENTER**, the **Event-ID** is protected. After the information on the **Specify Job Request Events** panel has been successfully entered, return to the **Add Job Request** panel by entering the **END** command (we will use **PF3**).

```
05/01/26 15:00 ------  PJS - SPECIFY JOB REQUEST EVENTS  --- LINE 1    OF 3
COMMAND ===> ____________________________________________________ SCROLL ===> PAGE

Request-ID   ===> TCHTST.EX5JOB6       Status       ===> WAIT
Description  ===> Example 5 - Job 6 - Using Advanced Events

S      Event-ID     Prepost  Date/Time Posted
_  TCHTST.JOB3END     YES
_  TCHTST.JOB4END     YES
_  TCHTST.JOB5END     YES
_  ________________   ___
_  ________________   ___
_  ________________   ___
_  ________________   ___
_  ________________   ___
_  ________________   ___
_  ________________   ___

Commands ==> JCL - JCL Source, FREQ - Frequency, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
Line Commands ==> D - Delete Event, P - Post Job Event, R - Reset Job Event
```

Press **PF3**

We have now entered all the parameters we need for this job request. The **Add Job Request** panel has been updated to show the parameters we have entered. We now complete the add and return to the **Job Request Menu** panel by entering the **END** command (once again, we use **PF3**).

```
05/01/26 15:00 -------  PJS - ADD JOB REQUEST  --------------------------------
COMMAND ===> ____________________________________________________________________

Request-ID   ===> TCHTST.EX5JOB6
Description  ===> Example 5 - Job 6 - Using Advanced Events________

Status       ===> WAIT                 Enabled      ===> YES  (Yes or No)

JCL Data Set ===> TCHTST.MISC.JCL(JOB6)

Next Run D/T ===> 01/26/2005 22:00
Window       ===>

Start D/T    ===> 01/26/2005 22:00     End D/T      ===>

Frequency    ===> ONCE

Events       ===> 00 OF 03

Commands ==> JCL - JCL Source, FREQ - Frequency, EVENT - Events,
             OPT - Submit Options, VAR - Variables,
             END - Complete Add, CANCEL - Cancel Add
```

Press **PF3**

The job request add for JOB6 is now complete. PJS will submit this job every day after JOB3, JOB4, and JOB5 have all completed.

```
05/01/26 15:00 -------  PJS - JOB REQUEST MENU  ------------ Job request added
OPTION ===> ____________________________________________________________________

Select one of the following functions:

  L  LIST     - List all Job Requests for Owner
  A  ADD      - Add a new Job Request
  D  DELETE   - Delete a Job Request
  M  MODIFY   - Modify a Job Request
  S  DISPLAY  - Display a Job Request


Request-ID:
  Owner-ID     ===> TCHTST__
  Request Name ===> EX5JOB6_   (Required for options D, M, and S)


Enter Request-ID (if required) and make a selection, then press ENTER.
```

Press **PF3**

Example 6: How to Run a Job After CICS Terminates

Problem

In this example, suppose you want to run a clean-up job any time CICS terminates. The time of day is not important. Assume that the JCL for the jobs is in a PDS named '**TCHTST.MISC.JCL**', in the member named **CLEANUP.**

To schedule this job, we will use an event that we will name **CICSEND**. Since this event may be of use to many users, we will assign it an Owner-ID of **CICSGRP**. The JCL for CICS will be modified to post this event as the last step (even if CICS abends). The clean-up job will be scheduled with a frequency of 1 minute, and specifying that the event must be posted before the job is run. Thus, the job will always be ready to run as soon as the event is posted.

Before defining the job request to PJS, modify the CICS JCL to include the PJSPOST step at the end of the job as follows:

```
//CICS     JOB
//  existing JCL statements
//            .
//            .
//            .
//PJSPOST EXEC PGM=PJSPOST,PARM='CICSGRP.CICSEND',
//             COND=EVEN
```

The **PARM** parameter specifies the name of the event to be posted. The **COND** parameter is used to post the event even if the job terminates abnormally.

TSO Solution

Use the **PJREQADD** command to add the job request as follows:

```
 READY
pjreqadd id(example6)
 REQ ADD
set desc('Example 6 - Run a Job After CICS Terminates')
 REQ ADD
set jcldsn('tchtst.misc.jcl(cleanup)')
 REQ ADD
set minute(1) strtdate(*) strttime(*)
 REQ ADD
event add cicsgrp.cicsend
 REQ ADD
end
 PJS2101I  Job request TCHTST.EXAMPLE6 successfully added
 READY
pjreqlst id(example6)
    Request-ID          Description
 TCHTST.EXAMPLE6    Example 6 - Run a Job After CICS Terminates

                         Next Run        Window        Notify    Notify
           Status      Date    Time   Time  Option    Userid    Level
          WAIT      01/25/2005 14:45                 TCHTST    INFO

                   JCL Data Set Name                                Save
          TCHTST.MISC.JCL(CLEANUP)                                 NOSAVE

             Last Submit          Status Change      Spool      Spool    Retry
            Date      Time       Date      Time      Member   Rec Count  Count
                                01/25/2005 14:44:46                    0      0

          Userid   Group
          TCHTST   WTST

                                          Start              End
              Frequency              Date    Time      Date    Time
            1 MINUTES            01/25/2005 14:44

                                Pre-        Posted
              Event-ID          Post    Date      Time
          CICSGRP.CICSEND       YES

 READY
```

ISPF Solution

Use the **Add Job Request** dialog to add the job request as follows:

Access the **PJS Main Menu** panel from your **ISPF Primary Option Menu** panel, or by what ever other means your installation uses. Then select option **J** to access the **PJS Job Request Menu** panel.

```
 05/01/26 15:06 -------  PJS - PERSONAL JOB SCHEDULER  ---------------- 3.1.0
 OPTION ===> J_____________________________________________________________

Select one of the following functions:

  J  JOBREQ   - Update Job Requests
  C  CALENDAR - Update Calendars
  E  EVENT    - Update Events
  V  VARIABLE - Update Variables

  T  TUTORIAL - Enter PJS/ISPF Tutorial


Select an option, then press ENTER.


(c) Copyright, Northrop Grumman, 1990, 2005.
For additional copyright information enter ABOUT on the command line.

PJS is distributed under the GNU General Public License.
For license information enter LICENSE on the command line.
```

Press **ENTER**

On the **PJS Job Request Menu** panel, enter the **Request Name**, then select option **A** to access the **Add Job Request** panel. The **Owner-ID** defaults to the TSO User-ID.

```
05/01/26 15:06 -------  PJS - JOB REQUEST MENU  -------------------------------
OPTION ===> A____________________________________________________________________

Select one of the following functions:

  L  LIST     - List all Job Requests for Owner
  A  ADD      - Add a new Job Request
  D  DELETE   - Delete a Job Request
  M  MODIFY   - Modify a Job Request
  S  DISPLAY  - Display a Job Request


Request-ID:
  Owner-ID     ===> TCHTST__
  Request Name ===> EXAMPLE6   (Required for options D, M, and S)


Enter Request-ID (if required) and make a selection, then press ENTER.
```

Press **ENTER**

From the **Add Job Request** panel, enter the **Request Description**, then enter the **JCL** command to access the **Specify Job Request JCL Source** panel.

```
05/01/26 15:06 -------  PJS - ADD JOB REQUEST  --------------------------------
COMMAND ===> JCL_____________________________________________________________

Request-ID   ===> TCHTST.EXAMPLE6
Description  ===> Example 6 - Run a Job After CICS Terminates_______

Status       ===> WAIT                  Enabled      ===> YES  (Yes or No)

JCL Data Set ===>

Next Run D/T ===>
Window       ===>

Start D/T    ===>                       End D/T      ===>

Frequency    ===> ONCE

Events       ===> NONE

Commands ==> JCL - JCL Source, FREQ - Frequency, EVENT - Events,
             OPT - Submit Options, VAR - Variables,
             END - Complete Add, CANCEL - Cancel Add
```

Press **ENTER**

On the **Specify Job Request JCL Source** panel, enter the **JCL Data Set Name** and the **Member Name** in which the JCL has been placed. The **JCL Save Option** defaults to **NO**.

```
05/01/26 15:06 -------  PJS - SPECIFY JCL SOURCE  ----------------------------
COMMAND ===> ____________________________________________________________________

Request-ID   ===> TCHTST.EXAMPLE6      Status       ===> WAIT
Description  ===>  Example 6 - Run a Job After CICS Terminates


JCL Data Set Name ===> TCHTST.MISC.JCL_____________________________
    Member Name   ===> CLEANUP_

JCL Save Option   ===> NO_____      (Yes, No, or Refresh)





Commands ==> BROWSE - Browse JCL, BRDATA - Browse JCL Data Set,
             FREQ - Frequency, EVENT - Events, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
```

Press **ENTER**

After the information on the **Specify Job Request JCL Source** panel has been successfully entered, enter the **FREQ** command to access the **Specify Job Request Frequency** panel.

```
05/01/26 15:06 -------  PJS - SPECIFY JCL SOURCE  ----------------------------
COMMAND ===> FREQ_____________________________________________________________

Request-ID   ===> TCHTST.EXAMPLE6      Status       ===> WAIT
Description  ===> Example 6 - Run a Job After CICS Terminates


JCL Data Set Name ===> TCHTST.MISC.JCL_____________________________
    Member Name   ===> CLEANUP_

JCL Save Option   ===> NO_____      (Yes, No, or Refresh)





Commands ==> BROWSE - Browse JCL, BRDATA - Browse JCL Data Set,
             FREQ - Frequency, EVENT - Events, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
```

Press **ENTER**

On the **Specify Job Request Frequency** panel, enter the **Start Date and Time**. In this case, the **Start Date** id entered as a **relative date**, and the **Start Time** is entered as a **relative time** to specify 'now'. The **Frequency** is entered as a **Periodic** frequency by entering **1** in the number field, and **MINUTES** in the units field.

```
05/01/26 15:06 -------  PJS - SPECIFY JOB FREQUENCY  --------------------------
COMMAND ===> ______________________________________________________________________

Request-ID   ===> TCHTST.EXAMPLE6      Status       ===> WAIT
Description  ===> Example 6 - Run a Job After CICS Terminates

Next Run D/T ===>                      Recalculate  ===> ___  (Yes or No)

Start D/T    ===> *________ *_______   End D/T      ===> _________ ________

Frequency (choose one of the options below)
   Periodic     ===> 1__ (num) minutes (units - Yr, Mo, Wk, Day, Hr, or Min)
      or
   Day of Week  ===> _ Sun  _ Mon  _ Tue  _ Wed  _ Thu  _ Fri  _ Sat
      or
   End of Month ===> Last Day - __   (days before last day of each month)
      or
   Calendar(s)  ===> _________________ _________________ _________________
      or
   Once (if none of the above is specified)

Commands ==> JCL - JCL Source, EVENT - Events, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
```

Press **ENTER**

After pressing **ENTER**, the **Next Run Date and Time** is calculated. Also note that the **relative date** and the **relative time** we entered for the **Start Date and Time** has been translated to an **absolute date** and an **absolute time**.

After the information on the **Specify Job Request Frequency** panel has been successfully entered, enter the **EVENT** command to access the **Specify Job Request Events** panel.

```
05/01/26 15:06 -------  PJS - SPECIFY JOB FREQUENCY  --------------------------
COMMAND ===> EVENT____________________________________________________________

Request-ID   ===> TCHTST.EXAMPLE6      Status       ===> WAIT
Description  ===> Example 6 - Run a Job After CICS Terminates

Next Run D/T ===> 01/26/2005 15:07     Recalculate  ===> ___  (Yes or No)

Start D/T    ===> 01/26/2005 15:06     End D/T      ===> _________ ________

Frequency (choose one of the options below)
   Periodic     ===> 001 (num) MINUTES (units - Yr, Mo, Wk, Day, Hr, or Min)
      or
   Day of Week  ===> _ Sun  _ Mon  _ Tue  _ Wed  _ Thu  _ Fri  _ Sat
      or
   End of Month ===> Last Day - __   (days before last day of each month)
      or
   Calendar(s)  ===> _________________ _________________ _________________
      or
   Once (if none of the above is specified)

Commands ==> JCL - JCL Source, EVENT - Events, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
```

Press **PF3**

On the **Specify Job Request Events** panel, enter the **Event-ID** of the event on which this job depends (**CICSGRP.CICSEND**).

```
05/01/26 15:06 ------  PJS - SPECIFY JOB REQUEST EVENTS  --- LINE 1    OF 0
COMMAND ===> ____________________________________________________ SCROLL ===> PAGE

Request-ID   ===> TCHTST.EXAMPLE6      Status       ===> WAIT
Description  ===> Example 6 - Run a Job After CICS Terminates

S      Event-ID     Prepost  Date/Time Posted
_  cicsgrp.cicsend__  ___
_  ________________  ___
_  ________________  ___
_  ________________  ___
_  ________________  ___
_  ________________  ___
_  ________________  ___
_  ________________  ___
_  ________________  ___
_  ________________  ___

Commands ==> JCL - JCL Source, FREQ - Frequency, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
Line Commands ==> D - Delete Event, P - Post Job Event, R - Reset Job Event
```

Press **ENTER**

After pressing **ENTER**, the **Event-ID** is protected. After the information on the **Specify Job Request Events** panel has been successfully entered, return to the **Add Job Request** panel by entering the **END** command (we will use **PF3**).

```
05/01/26 15:06 ------  PJS - SPECIFY JOB REQUEST EVENTS  --- LINE 1    OF 1
COMMAND ===> ____________________________________________________ SCROLL ===> PAGE

Request-ID   ===> TCHTST.EXAMPLE6      Status       ===> WAIT
Description  ===> Example 6 - Run a Job After CICS Terminates

S      Event-ID     Prepost  Date/Time Posted
_  CICSGRP.CICSEND    YES
_  ________________   ___
_  ________________   ___
_  ________________   ___
_  ________________   ___
_  ________________   ___
_  ________________   ___
_  ________________   ___
_  ________________   ___
_  ________________   ___

Commands ==> JCL - JCL Source, FREQ - Frequency, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
Line Commands ==> D - Delete Event, P - Post Job Event, R - Reset Job Event
```

Press **PF3**

We have now entered all the parameters we need for this job request. The **Add Job Request** panel has been updated to show the parameters we have entered. We now complete the add and return to the **Job Request Menu** panel by entering the **END** command (once again, we use **PF3**).

```
05/01/26 15:06 -------  PJS - ADD JOB REQUEST  --------------------------------
COMMAND ===> ____________________________________________________________________

Request-ID   ===> TCHTST.EXAMPLE6      Status       ===> WAIT
Description  ===> Example 6 - Run a Job After CICS Terminates_______

Status       ===> WAIT                 Enabled      ===> YES  (Yes or No)

JCL Data Set ===> TCHTST.MISC.JCL(CLEANUP)

Next Run D/T ===> 01/26/2005 15:07
Window       ===>

Start D/T    ===> 01/26/2005 15:06     End D/T      ===>

Frequency    ===> 001 MINUTES

Events       ===> 00 OF 01

Commands ==> JCL - JCL Source, FREQ - Frequency, EVENT - Events,
             OPT - Submit Options, VAR - Variables,
             END - Complete Add, CANCEL - Cancel Add
```

Press **PF3**

The job request add is now complete. PJS will submit this job whenever CICS terminates.

```
05/01/26 15:06 -------  PJS - JOB REQUEST MENU  ------------ Job request added
OPTION ===> ____________________________________________________________________

Select one of the following functions:

  L  LIST     - List all Job Requests for Owner
  A  ADD      - Add a new Job Request
  D  DELETE   - Delete a Job Request
  M  MODIFY   - Modify a Job Request
  S  DISPLAY  - Display a Job Request


Request-ID:
  Owner-ID     ===> TCHTST__
  Request Name ===> EXAMPLE6   (Required for options D, M, and S)


Enter Request-ID (if required) and make a selection, then press ENTER.
```

Press **PF3**

Example 7: How to Run a Weekly Job After a Daily Job

Problem

In this example, suppose you want to run two jobs. The first job is to be submitted every wekday (Monday through Friday) at 6:00 PM. The second job is to be submitted every Friday after the first successfully completes. Assume that the JCL for the jobs is in a PDS named '**TCHTST.MISC.JCL**'. The first job is in the member named **JOB1**. The second job is in the member named **JOB2**.

To schedule these jobs, we will use an event that we will name **JOB1END**. The JCL for JOB1 will be modified to post this event as the last step, if the previous steps have completed successfully. JOB1 will be scheduled with a simple job request, as shown in Example 1. JOB2 will be scheduled with a job request that specifies a run time the same as JOB1, but will also specify that the event must be posted before the job is run.

This is similar to the situation in Example 4, but in this case, the second job is not run every time the first job is run. If The event were specified as in Example 4, the event wound be posted when JOB1 ran on Monday. When JOB2 is scheduled to run on Friday, the job request event will still be posted (from Monday), and the job may run before JOB1 completes.

To prevent this from happening, the job request event is specified with the **NOPREPOST** option. This prevents the job request event from being posted before JOB2 is scheduled to run on Friday. the job request event will be posted by Friday's JOB1, and JOB2 will wait on this posting.

Before defining the job requests to PJS, first place the JCL in the specified data set and members. Modify JOB1 to include the PJSPOST step at the end of the job as follows:

```
//JOB1     JOB
//  existing JCL statements
//            .
//            .
//            .
//PJSPOST EXEC PGM=PJSPOST,PARM='TCHTST.JOB1END',
//             COND=(0,NE)
```

The **PARM** parameter specifies the name of the event to be posted. The **COND** parameter is used to post the event only if all previous job steps completed with zero return codes.

TSO Solution

Use the **PJREQADD** command to add the job request for JOB1 as follows:

```
 READY
pjreqadd id(ex7job1)
 REQ ADD
set desc('Example 7 - Job 1 - Using NOPREPOST Events')
 REQ ADD
set jcldsn('tchtst.misc.jcl(job1)')
 REQ ADD
set weekdays(mon,tue,wed,thu,fri)
 REQ ADD
set strtdate(*) strttime(6:00pm)
 REQ ADD
end
 PJS2101I  Job request TCHTST.EX7JOB1 successfully added
 READY
pjreqlst id(ex7job1)
    Request-ID          Description
 TCHTST.EX7JOB1     Example 7 - Job 1 - Using NOPREPOST Events

                         Next Run        Window        Notify    Notify
           Status      Date    Time   Time  Option    Userid    Level
          WAIT      01/25/2005 18:00                 TCHTST    INFO

                    JCL Data Set Name                                Save
          TCHTST.MISC.JCL(JOB1)                                     NOSAVE

             Last Submit          Status Change      Spool      Spool    Retry
            Date      Time       Date      Time      Member   Rec Count  Count
                                01/25/2005 14:49:54                    0      0

          Userid   Group
          TCHTST   WTST

                                      Start              End
              Frequency             Date    Time      Date    Time
          MO,TU,WE,TH,FR         01/25/2005 18:00

 READY
```

Next, use the **PJREQADD** command to add the job request for JOB2 as follows:

```
 READY
pjreqadd id(ex7job2)
 REQ ADD
set desc('Example 7 - Job 2 - Using NOPREPOST Events')
 REQ ADD
set jcldsn('tchtst.misc.jcl(job2)')
 REQ ADD
set weekdays(fri)
 REQ ADD
set strtdate(*) strttime(6:00pm)
 REQ ADD
event add job1end noprepost
 REQ ADD
end
 PJS2101I  Job request TCHTST.EX7JOB2 successfully added
 READY
pjreqlst id(ex7job2)
    Request-ID          Description
 TCHTST.EX7JOB2     Example 7 - Job 2 - Using NOPREPOST Events

                         Next Run        Window        Notify    Notify
           Status      Date    Time   Time  Option    Userid    Level
          WAIT      01/28/2005 18:00                 TCHTST    INFO

                   JCL Data Set Name                                Save
          TCHTST.MISC.JCL(JOB2)                                    NOSAVE

             Last Submit          Status Change      Spool      Spool    Retry
            Date      Time       Date      Time      Member   Rec Count  Count
                                01/25/2005 14:53:31                    0      0

          Userid   Group
          TCHTST   WTST

                                         Start              End
              Frequency             Date    Time      Date    Time
          FR                     01/25/2005 18:00

                              Pre-        Posted
              Event-ID        Post    Date      Time
          TCHTST.JOB1END      NO

 READY
```

ISPF Solution

Use the **Add Job Request** dialog to add the job request for the first job as follows:

Access the **PJS Main Menu** panel from your **ISPF Primary Option Menu** panel, or by what ever other means your installation uses. Then select option **J** to access the **PJS Job Request Menu** panel.

```
 05/01/26 15:49 -------  PJS - PERSONAL JOB SCHEDULER  ---------------- 3.1.0
 OPTION ===> J_____________________________________________________________

Select one of the following functions:

  J  JOBREQ   - Update Job Requests
  C  CALENDAR - Update Calendars
  E  EVENT    - Update Events
  V  VARIABLE - Update Variables

  T  TUTORIAL - Enter PJS/ISPF Tutorial


Select an option, then press ENTER.


(c) Copyright, Northrop Grumman, 1990, 2005.
For additional copyright information enter ABOUT on the command line.

PJS is distributed under the GNU General Public License.
For license information enter LICENSE on the command line.
```

Press **ENTER**

On the **PJS Job Request Menu** panel, enter the **Request Name**, then select option **A** to access the **Add Job Request** panel. The **Owner-ID** defaults to the TSO User-ID.

```
05/01/26 15:49 -------  PJS - JOB REQUEST MENU  -------------------------------
OPTION ===> A____________________________________________________________________

Select one of the following functions:

  L  LIST     - List all Job Requests for Owner
  A  ADD      - Add a new Job Request
  D  DELETE   - Delete a Job Request
  M  MODIFY   - Modify a Job Request
  S  DISPLAY  - Display a Job Request


Request-ID:
  Owner-ID     ===> TCHTST__
  Request Name ===> EX7JOB1_   (Required for options D, M, and S)


Enter Request-ID (if required) and make a selection, then press ENTER.
```

Press **ENTER**

From the **Add Job Request** panel, enter the **Request Description**, then enter the **JCL** command to access the **Specify Job Request JCL Source** panel.

```
05/01/26 15:49 -------  PJS - ADD JOB REQUEST  --------------------------------
COMMAND ===> JCL_____________________________________________________________

Request-ID   ===> TCHTST.EX7JOB1
Description  ===> Example 7 - Job 1 - Using NOPREPOST Events________

Status       ===> WAIT                  Enabled      ===> YES  (Yes or No)

JCL Data Set ===>

Next Run D/T ===>
Window       ===>

Start D/T    ===>                       End D/T      ===>

Frequency    ===> ONCE

Events       ===> NONE

Commands ==> JCL - JCL Source, FREQ - Frequency, EVENT - Events,
             OPT - Submit Options, VAR - Variables,
             END - Complete Add, CANCEL - Cancel Add
```

Press **ENTER**

On the **Specify Job Request JCL Source** panel, enter the **JCL Data Set Name** and the **Member Name** in which the JCL has been placed. The **JCL Save Option** defaults to **NO**.

```
05/01/26 15:49 -------  PJS - SPECIFY JCL SOURCE  ----------------------------
COMMAND ===> ____________________________________________________________________

Request-ID   ===> TCHTST.EX7JOB1       Status       ===> WAIT
Description  ===> Example 7 - Job 1 - Using NOPREPOST Events


JCL Data Set Name ===> TCHTST.MISC.JCL_____________________________
    Member Name   ===> JOB1____

JCL Save Option   ===> NO_____      (Yes, No, or Refresh)





Commands ==> BROWSE - Browse JCL, BRDATA - Browse JCL Data Set,
             FREQ - Frequency, EVENT - Events, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
```

Press **ENTER**

After the information on the **Specify Job Request JCL Source** panel has been successfully entered, enter the **FREQ** command to access the **Specify Job Request Frequency** panel.

```
05/01/26 15:49 -------  PJS - SPECIFY JCL SOURCE  ----------------------------
COMMAND ===> FREQ_____________________________________________________________

Request-ID   ===> TCHTST.EX7JOB1       Status       ===> WAIT
Description  ===> Example 7 - Job 1 - Using NOPREPOST Events


JCL Data Set Name ===> TCHTST.MISC.JCL_____________________________
    Member Name   ===> JOB1____

JCL Save Option   ===> NO_____      (Yes, No, or Refresh)





Commands ==> BROWSE - Browse JCL, BRDATA - Browse JCL Data Set,
             FREQ - Frequency, EVENT - Events, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
```

Press **ENTER**

On the **Specify Job Request Frequency** panel, enter the **Start Date and Time**. In this case, the **Start Date** is entered as a **relative date**, to specify 'today'. The Start Time is entered as an **absolute time** to specify the specific time at which the job is to be run. The **Frequency** is entered as a **Day-of-Week** frequency by entering a non-blank character in the unprotected field immediately before the weekdays to be selected.

```
05/01/26 15:49 -------  PJS - SPECIFY JOB FREQUENCY  -------------------------
COMMAND ===> ____________________________________________________________________

Request-ID   ===> TCHTST.EX7JOB1       Status       ===> WAIT
Description  ===> Example 7 - Job 1 - Using NOPREPOST Events

Next Run D/T ===>                      Recalculate  ===> ___  (Yes or No)

Start D/T    ===> *________ 06:00 pm  End D/T      ===> _________ ________

Frequency (choose one of the options below)
   Periodic     ===> ___ (num) _______ (units - Yr, Mo, Wk, Day, Hr, or Min)
      or
   Day of Week  ===> _ Sun  s Mon  s Tue  s Wed  s Thu  s Fri  _ Sat
      or
   End of Month ===> Last Day - __   (days before last day of each month)
      or
   Calendar(s)  ===> _________________ _________________ _________________
      or
   Once (if none of the above is specified)

Commands ==> JCL - JCL Source, EVENT - Events, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
```

Press **ENTER**

After pressing **ENTER**, the **Next Run Date and Time** is calculated. Also note that the **relative date** we entered for the **Start Date** has been translated to an **absolute date**. After the information on the **Specify Job Request JCL Source** panel has been successfully entered, return to the **Add Job Request** panel by entering the **END** command (we will use **PF3**).

```
05/01/26 15:49 -------  PJS - SPECIFY JOB FREQUENCY  -------------------------
COMMAND ===> ____________________________________________________________________

Request-ID   ===> TCHTST.EX7JOB1       Status       ===> WAIT
Description  ===> Example 7 - Job 1 - Using NOPREPOST Events

Next Run D/T ===> 01/26/2005 18:00     Recalculate  ===> ___  (Yes or No)

Start D/T    ===> 01/26/2005 18:00     End D/T      ===> _________ ________

Frequency (choose one of the options below)
   Periodic     ===> ___ (num) _______ (units - Yr, Mo, Wk, Day, Hr, or Min)
      or
   Day of Week  ===> _ Sun  S Mon  S Tue  S Wed  S Thu  S Fri  _ Sat
      or
   End of Month ===> Last Day - __   (days before last day of each month)
      or
   Calendar(s)  ===> _________________ _________________ _________________
      or
   Once (if none of the above is specified)

Commands ==> JCL - JCL Source, EVENT - Events, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
```

Press **PF3**

We have now entered all the parameters we need for this job request. The **Add Job Request** panel has been updated to show the parameters we have entered. We now complete the add and return to the **Job Request Menu** panel by entering the **END** command (once again, we use **PF3**).

```
05/01/26 15:49 -------  PJS - ADD JOB REQUEST  --------------------------------
COMMAND ===> ____________________________________________________________________

Request-ID   ===> TCHTST.EX7JOB1
Description  ===> Example 7 - Job 1 - Using NOPREPOST Events________

Status       ===> WAIT                 Enabled      ===> YES  (Yes or No)

JCL Data Set ===> TCHTST.MISC.JCL(JOB1)

Next Run D/T ===> 01/26/2005 18:00
Window       ===>

Start D/T    ===> 01/26/2005 18:00     End D/T      ===>

Frequency    ===> MO,TU,WE,TH,FR

Events       ===> NONE

Commands ==> JCL - JCL Source, FREQ - Frequency, EVENT - Events,
             OPT - Submit Options, VAR - Variables,
             END - Complete Add, CANCEL - Cancel Add
```

Press **PF3**

The first job request add is now complete. PJS will submit this at 6:00 PM today.

Next, use the **Add Job Request** dialog to add the job request for the second job as follows:

Enter the **Request Name** for the second job request, then select option **A** to access the **Add Job Request** panel. The **Owner-ID,** once again, defaults to the TSO User-ID.

```
05/01/26 15:50 -------  PJS - JOB REQUEST MENU  ------------ Job request added
OPTION ===> A

Select one of the following functions:

  L  LIST     - List all Job Requests for Owner
  A  ADD      - Add a new Job Request
  D  DELETE   - Delete a Job Request
  M  MODIFY   - Modify a Job Request
  S  DISPLAY  - Display a Job Request


Request-ID:
  Owner-ID     ===> TCHTST
  Request Name ===> EX7JOB2   (Required for options D, M, and S)


Enter Request-ID (if required) and make a selection, then press ENTER.
```

Press **PF3**

From the **Add Job Request** panel, enter the **Request Description**, then enter the **JCL** command to access the **Specify Job Request JCL Source** panel.

```
05/01/26 15:50 -------  PJS - ADD JOB REQUEST  --------------------------------
COMMAND ===> JCL_____________________________________________________________

Request-ID   ===> TCHTST.EX7JOB2
Description  ===> Example 7 - Job 2 - Using NOPREPOST Events________

Status       ===> WAIT                 Enabled      ===> YES  (Yes or No)

JCL Data Set ===>

Next Run D/T ===>
Window       ===>

Start D/T    ===>                      End D/T      ===>

Frequency    ===> ONCE

Events       ===> NONE

Commands ==> JCL - JCL Source, FREQ - Frequency, EVENT - Events,
             OPT - Submit Options, VAR - Variables,
             END - Complete Add, CANCEL - Cancel Add
```

Press **ENTER**

On the **Specify Job Request JCL Source** panel, enter the **JCL Data Set Name** and the **Member Name** in which the JCL has been placed. The **JCL Save Option** defaults to **NO**.

```
05/01/26 15:51 -------  PJS - SPECIFY JCL SOURCE  ----------------------------
COMMAND ===> ____________________________________________________________________

Request-ID   ===> TCHTST.EX7JOB2       Status       ===> WAIT
Description  ===> Example 7 - Job 2 - Using NOPREPOST Events


JCL Data Set Name ===> TCHTST.MISC.JCL_____________________________
    Member Name   ===> JOB2____

JCL Save Option   ===> NO_____      (Yes, No, or Refresh)





Commands ==> BROWSE - Browse JCL, BRDATA - Browse JCL Data Set,
             FREQ - Frequency, EVENT - Events, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
```

Press **ENTER**

After the information on the **Specify Job Request JCL Source** panel has been successfully entered, enter the **FREQ** command to access the **Specify Job Request Frequency** panel.

```
05/01/26 15:51 -------  PJS - SPECIFY JCL SOURCE  ----------------------------
COMMAND ===> FREQ_____________________________________________________________

Request-ID   ===> TCHTST.EX7JOB2       Status       ===> WAIT
Description  ===> Example 7 - Job 2 - Using NOPREPOST Events


JCL Data Set Name ===> TCHTST.MISC.JCL_____________________________
    Member Name   ===> JOB2____

JCL Save Option   ===> NO_____      (Yes, No, or Refresh)





Commands ==> BROWSE - Browse JCL, BRDATA - Browse JCL Data Set,
             FREQ - Frequency, EVENT - Events, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
```

Press **ENTER**

On the **Specify Job Request Frequency** panel, enter the **Start Date and Time**. In this case, the **Start Date** is entered as a **relative date**, to specify 'today'. The Start Time is entered as an **absolute time** to specify the specific time at which the job is to be run. The **Frequency** is entered as a **Day-of-Week** frequency by entering a non-blank character in the unprotected field immediately before the weekday to be selected.

```
05/01/26 15:51 -------  PJS - SPECIFY JOB FREQUENCY  --------------------------
COMMAND ===> ____________________________________________________________________

Request-ID   ===> TCHTST.EX7JOB2       Status       ===> WAIT
Description  ===> Example 7 - Job 2 - Using NOPREPOST Events

Next Run D/T ===>                      Recalculate  ===> ___  (Yes or No)

Start D/T    ===> *________ 06:00 pm   End D/T      ===> _________ ________

Frequency (choose one of the options below)
   Periodic     ===> ___ (num) _______ (units - Yr, Mo, Wk, Day, Hr, or Min)
      or
   Day of Week  ===> _ Sun  _ Mon  _ Tue  _ Wed  _ Thu  s Fri  _ Sat
      or
   End of Month ===> Last Day - __   (days before last day of each month)
      or
   Calendar(s)  ===> _________________ _________________ _________________
      or
   Once (if none of the above is specified)

Commands ==> JCL - JCL Source, EVENT - Events, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
```

Press **ENTER**

After pressing **ENTER**, the **Next Run Date and Time** is calculated. Also note that the **relative date** we entered for the **Start Date** has been translated to an **absolute date**.

After the information on the **Specify Job Request Frequency** panel has been successfully entered, enter the **EVENT** command to access the **Specify Job Request Events** panel.

```
05/01/26 15:51 -------  PJS - SPECIFY JOB FREQUENCY  --------------------------
COMMAND ===> EVENT____________________________________________________________

Request-ID   ===> TCHTST.EX7JOB2       Status       ===> WAIT
Description  ===> Example 7 - Job 2 - Using NOPREPOST Events

Next Run D/T ===> 01/28/2005 18:00     Recalculate  ===> ___  (Yes or No)

Start D/T    ===> 01/26/2005 18:00     End D/T      ===> _________ ________

Frequency (choose one of the options below)
   Periodic     ===> ___ (num) _______ (units - Yr, Mo, Wk, Day, Hr, or Min)
      or
   Day of Week  ===> _ Sun  _ Mon  _ Tue  _ Wed  _ Thu  S Fri  _ Sat
      or
   End of Month ===> Last Day - __   (days before last day of each month)
      or
   Calendar(s)  ===> _________________ _________________ _________________
      or
   Once (if none of the above is specified)

Commands ==> JCL - JCL Source, EVENT - Events, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
```

Press **ENTER**

On the **Specify Job Request Events** panel, enter the **Event-ID** of the event on which this job depends (**JOB1END**).

```
05/01/26 15:51 ------  PJS - SPECIFY JOB REQUEST EVENTS  --- LINE 1    OF 0
COMMAND ===> ____________________________________________________ SCROLL ===> PAGE

Request-ID   ===> TCHTST.EX7JOB2       Status       ===> WAIT
Description  ===> Example 7 - Job 2 - Using NOPREPOST Events

S      Event-ID     Prepost  Date/Time Posted
_  job1end__________  no_
_  ________________  ___
_  ________________  ___
_  ________________  ___
_  ________________  ___
_  ________________  ___
_  ________________  ___
_  ________________  ___
_  ________________  ___
_  ________________  ___

Commands ==> JCL - JCL Source, FREQ - Frequency, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
Line Commands ==> D - Delete Event, P - Post Job Event, R - Reset Job Event
```

Press **ENTER**

After pressing **ENTER**, the **Event-ID** is protected. After the information on the **Specify Job Request Events** panel has been successfully entered, return to the **Add Job Request** panel by entering the **END** command (we will use **PF3**).

```
05/01/26 15:52 ------  PJS - SPECIFY JOB REQUEST EVENTS  --- LINE 1    OF 1
COMMAND ===> ____________________________________________________ SCROLL ===> PAGE

Request-ID   ===> TCHTST.EX7JOB2       Status       ===> WAIT
Description  ===> Example 7 - Job 2 - Using NOPREPOST Events

S      Event-ID     Prepost  Date/Time Posted
_  TCHTST.JOB1END     NO_
_  ________________   ___
_  ________________   ___
_  ________________   ___
_  ________________   ___
_  ________________   ___
_  ________________   ___
_  ________________   ___
_  ________________   ___
_  ________________   ___
_  ________________   ___

Commands ==> JCL - JCL Source, FREQ - Frequency, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
Line Commands ==> D - Delete Event, P - Post Job Event, R - Reset Job Event
```

Press **PF3**

We have now entered all the parameters we need for this job request. The **Add Job Request** panel has been updated to show the parameters we have entered. We now complete the add and return to the **Job Request Menu** panel by entering the **END** command (once again, we use **PF3**).

```
05/01/26 15:52 -------  PJS - ADD JOB REQUEST  --------------------------------
COMMAND ===> ____________________________________________________________________

Request-ID   ===> TCHTST.EX7JOB2
Description  ===> Example 7 - Job 2 - Using NOPREPOST Events________

Status       ===> WAIT                 Enabled      ===> YES  (Yes or No)

JCL Data Set ===> TCHTST.MISC.JCL(JOB2)

Next Run D/T ===> 01/28/2005 18:00
Window       ===>

Start D/T    ===> 01/26/2005 18:00     End D/T      ===>

Frequency    ===> FR

Events       ===> 00 OF 01

Commands ==> JCL - JCL Source, FREQ - Frequency, EVENT - Events,
             OPT - Submit Options, VAR - Variables,
             END - Complete Add, CANCEL - Cancel Add
```

Press **PF3**

The second job request add is now complete. PJS will submit this job every friday after the first job completes.

```
05/01/26 15:52 -------  PJS - JOB REQUEST MENU  ------------ Job request added
OPTION ===> ____________________________________________________________________

Select one of the following functions:

  L  LIST     - List all Job Requests for Owner
  A  ADD      - Add a new Job Request
  D  DELETE   - Delete a Job Request
  M  MODIFY   - Modify a Job Request
  S  DISPLAY  - Display a Job Request


Request-ID:
  Owner-ID     ===> TCHTST__
  Request Name ===> EX7JOB2_   (Required for options D, M, and S)


Enter Request-ID (if required) and make a selection, then press ENTER.
```

Press **PF3**

Example 8: How to Run a Job After a Manual Operation

Problem

In this example, suppose you want to run a long running job after a coworker leaves for the day. Assume that the JCL for the jobs is in a PDS named **'TCHTST.MISC.JCL'**, in the member named **JOB1**.

To allow the coworker to submit your job without your password, we will use an event that we will name **USER02.IMGONE** (assuming USER02 is your coworkers TSO User-ID). Your job will be scheduled to run after 5:00 PM when the event is posted. The coworker will manually post the event when he/she leaves.

TSO Solution

Use the **PJREQADD** command to add the job request as follows:

```
 READY
pjreqadd id(example8)
 REQ ADD
set desc('Example 8 - Manually Posting an Event')
 REQ ADD
set jcldsn('tchtst.misc.jcl(job1)')
 REQ ADD
set strtdate(*) strttime(5:00pm)
 REQ ADD
event add user02.imgone
 REQ ADD
end
 PJS2101I  Job request TCHTST.EXAMPLE8 successfully added
 READY
pjreqlst id(example8)
    Request-ID          Description
 TCHTST.EXAMPLE8    Example 8 - Manually Posting an Event

                         Next Run        Window        Notify    Notify
           Status       Date    Time   Time  Option    Userid    Level
          WAIT      01/25/2005 17:00                 TCHTST    INFO

                   JCL Data Set Name                                Save
          TCHTST.MISC.JCL(JOB1)                                     NOSAVE

             Last Submit          Status Change      Spool      Spool    Retry
            Date      Time       Date      Time      Member   Rec Count  Count
                              01/25/2005 14:56:20                    0      0

          Userid   Group
          TCHTST   WTST

                                     Start              End
              Frequency            Date    Time      Date    Time
          ONCE                  01/25/2005 17:00

                            Pre-        Posted
              Event-ID      Post    Date      Time
          USER02.IMGONE     YES

 READY
```

When the coworker leaves for the day, he/she will use the **PJEVPOST** command to post the event as follows:

```
 READY
pjevpost imgone
     Event-ID       Last Post Date/Time     Status
 WSYS.IMGONE                            POST PENDING
 PJS2301I  Event WSYS.IMGONE  successfully posted
 READY
```

ISPF Solution

Use the **Add Job Request** dialog to add the job request for the first job as follows:

Access the **PJS Main Menu** panel from your **ISPF Primary Option Menu** panel, or by what ever other means your installation uses. Then select option **J** to access the **PJS Job Request Menu** panel.

```
 05/01/26 15:59 -------  PJS - PERSONAL JOB SCHEDULER  ---------------- 3.1.0
 OPTION ===> J_____________________________________________________________

Select one of the following functions:

  J  JOBREQ   - Update Job Requests
  C  CALENDAR - Update Calendars
  E  EVENT    - Update Events
  V  VARIABLE - Update Variables

  T  TUTORIAL - Enter PJS/ISPF Tutorial


Select an option, then press ENTER.


(c) Copyright, Northrop Grumman, 1990, 2005.
For additional copyright information enter ABOUT on the command line.

PJS is distributed under the GNU General Public License.
For license information enter LICENSE on the command line.
```

Press **ENTER**

Enter the **Request Name** for the job request, then select option **A** to access the **Add Job Request** panel. The **Owner-ID,** defaults to the TSO User-ID.

```
05/01/26 15:59 -------  PJS - JOB REQUEST MENU  -------------------------------
OPTION ===> A____________________________________________________________________

Select one of the following functions:

  L  LIST     - List all Job Requests for Owner
  A  ADD      - Add a new Job Request
  D  DELETE   - Delete a Job Request
  M  MODIFY   - Modify a Job Request
  S  DISPLAY  - Display a Job Request


Request-ID:
  Owner-ID     ===> TCHTST__
  Request Name ===> EXAMPLE8   (Required for options D, M, and S)


Enter Request-ID (if required) and make a selection, then press ENTER.
```

Press **PF3**

From the **Add Job Request** panel, enter the **Request Description**, then enter the **JCL** command to access the **Specify Job Request JCL Source** panel.

```
05/01/26 15:59 -------  PJS - ADD JOB REQUEST  --------------------------------
COMMAND ===> JCL_____________________________________________________________

Request-ID   ===> TCHTST.EXAMPLE8
Description  ===> Example 8 - Manually Posting an Event_____________

Status       ===> WAIT                  Enabled      ===> YES  (Yes or No)

JCL Data Set ===>

Next Run D/T ===>
Window       ===>

Start D/T    ===>                       End D/T      ===>

Frequency    ===> ONCE

Events       ===> NONE

Commands ==> JCL - JCL Source, FREQ - Frequency, EVENT - Events,
             OPT - Submit Options, VAR - Variables,
             END - Complete Add, CANCEL - Cancel Add
```

Press **ENTER**

On the **Specify Job Request JCL Source** panel, enter the **JCL Data Set Name** and the **Member Name** in which the JCL has been placed. The **JCL Save Option** defaults to **NO**.

```
05/01/26 15:59 -------  PJS - SPECIFY JCL SOURCE  ----------------------------
COMMAND ===> ____________________________________________________________________

Request-ID   ===> TCHTST.EXAMPLE8      Status       ===> WAIT
Description  ===> Example 8 - Manually Posting an Event


JCL Data Set Name ===> TCHTST.MISC.JCL_____________________________
    Member Name   ===> JOB1____

JCL Save Option   ===> NO_____      (Yes, No, or Refresh)





Commands ==> BROWSE - Browse JCL, BRDATA - Browse JCL Data Set,
             FREQ - Frequency, EVENT - Events, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
```

Press **ENTER**

After the information on the **Specify Job Request JCL Source** panel has been successfully entered, enter the **FREQ** command to access the **Specify Job Request Frequency** panel.

```
05/01/26 15:59 -------  PJS - SPECIFY JCL SOURCE  ----------------------------
COMMAND ===> FREQ_____________________________________________________________

Request-ID   ===> TCHTST.EXAMPLE8      Status       ===> WAIT
Description  ===> Example 8 - Manually Posting an Event


JCL Data Set Name ===> TCHTST.MISC.JCL_____________________________
    Member Name   ===> JOB1____

JCL Save Option   ===> NO_____      (Yes, No, or Refresh)





Commands ==> BROWSE - Browse JCL, BRDATA - Browse JCL Data Set,
             FREQ - Frequency, EVENT - Events, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
```

Press **ENTER**

On the **Specify Job Request Frequency** panel, enter the **Start Date and Time**. In this case, the **Start Date** is entered as a **relative date**, to specify 'today'. The **Start Time** is entered as an **absolute time** to specify the specific time at which the job is to be run. The **Frequency** is entered as a **Day-of-Week** frequency by entering a non-blank character in the unprotected field immediately before the weekday to be selected.

```
05/01/26 15:59 -------  PJS - SPECIFY JOB FREQUENCY  -------------------------
COMMAND ===> ______________________________________________________________________

Request-ID   ===> TCHTST.EXAMPLE8      Status       ===> WAIT
Description  ===> Example 8 - Manually Posting an Event

Next Run D/T ===>                      Recalculate  ===> ___  (Yes or No)

Start D/T    ===> *________ 05:00 pm   End D/T      ===> _________ ________

Frequency (choose one of the options below)
   Periodic     ===> ___ (num) _______ (units - Yr, Mo, Wk, Day, Hr, or Min)
      or
   Day of Week  ===> _ Sun  _ Mon  _ Tue  _ Wed  _ Thu  s Fri  _ Sat
      or
   End of Month ===> Last Day - __   (days before last day of each month)
      or
   Calendar(s)  ===> _________________ _________________ _________________
      or
   Once (if none of the above is specified)

Commands ==> JCL - JCL Source, EVENT - Events, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
```

Press **ENTER**

After pressing **ENTER**, the **Next Run Date and Time** is calculated. Also note that the **relative date** we entered for the **Start Date** has been translated to an **absolute date**.

After the information on the **Specify Job Request Frequency** panel has been successfully entered, enter the **EVENT** command to access the **Specify Job Request Events** panel.

```
05/01/26 15:59 -------  PJS - SPECIFY JOB FREQUENCY  --------------------------
COMMAND ===> EVENT_____________________________________________________________

Request-ID   ===> TCHTST.EXAMPLE8      Status       ===> WAIT
Description  ===> Example 8 - Manually Posting an Event

Next Run D/T ===> 01/26/2005 17:00     Recalculate  ===> ___  (Yes or No)

Start D/T    ===> 01/26/2005 17:00     End D/T      ===> _________ ________

Frequency (choose one of the options below)
   Periodic     ===> ___ (num) _______ (units - Yr, Mo, Wk, Day, Hr, or Min)
      or
   Day of Week  ===> _ Sun  _ Mon  _ Tue  _ Wed  _ Thu  S Fri  _ Sat
      or
   End of Month ===> Last Day - __   (days before last day of each month)
      or
   Calendar(s)  ===> _________________ _________________ _________________
      or
   Once (if none of the above is specified)

Commands ==> JCL - JCL Source, EVENT - Events, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
```

Press **ENTER**

On the **Specify Job Request Events** panel, enter the **Event-ID** of the event on which this job depends (**JOB1END**).

```
05/01/26 15:59 ------  PJS - SPECIFY JOB REQUEST EVENTS  --- LINE 1    OF 0
COMMAND ===> ____________________________________________________ SCROLL ===> PAGE

Request-ID   ===> TCHTST.EXAMPLE8      Status       ===> WAIT
Description  ===> Example 8 - Manually Posting an Event

S      Event-ID     Prepost  Date/Time Posted
_  user02.imgone____  ___
_  ________________  ___
_  ________________  ___
_  ________________  ___
_  ________________  ___
_  ________________  ___
_  ________________  ___
_  ________________  ___
_  ________________  ___
_  ________________  ___

Commands ==> JCL - JCL Source, FREQ - Frequency, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
Line Commands ==> D - Delete Event, P - Post Job Event, R - Reset Job Event
```

Press **ENTER**

After pressing **ENTER**, the **Event-ID** is protected. After the information on the **Specify Job Request Events** panel has been successfully entered, return to the **Add Job Request** panel by entering the **END** command (we will use **PF3**).

```
05/01/26 15:59 ------  PJS - SPECIFY JOB REQUEST EVENTS  --- LINE 1    OF 1
COMMAND ===> ____________________________________________________ SCROLL ===> PAGE

Request-ID   ===> TCHTST.EXAMPLE8      Status       ===> WAIT
Description  ===> Example 8 - Manually Posting an Event

S      Event-ID     Prepost  Date/Time Posted
_  USER02.IMGONE      YES
_  ________________   ___
_  ________________   ___
_  ________________   ___
_  ________________   ___
_  ________________   ___
_  ________________   ___
_  ________________   ___
_  ________________   ___
_  ________________   ___

Commands ==> JCL - JCL Source, FREQ - Frequency, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
Line Commands ==> D - Delete Event, P - Post Job Event, R - Reset Job Event
```

Press **PF3**

We have now entered all the parameters we need for this job request. The **Add Job Request** panel has been updated to show the parameters we have entered. We now complete the add and return to the **Job Request Menu** panel by entering the **END** command (once again, we use **PF3**).

```
05/01/26 16:00 -------  PJS - ADD JOB REQUEST  --------------------------------
COMMAND ===> ____________________________________________________________________

Request-ID   ===> TCHTST.EXAMPLE8
Description  ===> Example 8 - Manually Posting an Event_____________

Status       ===> WAIT                 Enabled      ===> YES  (Yes or No)

JCL Data Set ===> TCHTST.MISC.JCL(JOB1)

Next Run D/T ===> 01/26/2005 17:00
Window       ===>

Start D/T    ===> 01/26/2005 17:00     End D/T      ===>

Frequency    ===> ONCE

Events       ===> 00 OF 01

Commands ==> JCL - JCL Source, FREQ - Frequency, EVENT - Events,
             OPT - Submit Options, VAR - Variables,
             END - Complete Add, CANCEL - Cancel Add
```

Press **PF3**

The job request add is now complete. PJS will submit this job after 5:00 PM and after your coworker posts the event.

```
05/01/26 16:00 -------  PJS - JOB REQUEST MENU  ------------ Job request added
OPTION ===> ____________________________________________________________________

Select one of the following functions:

  L  LIST     - List all Job Requests for Owner
  A  ADD      - Add a new Job Request
  D  DELETE   - Delete a Job Request
  M  MODIFY   - Modify a Job Request
  S  DISPLAY  - Display a Job Request


Request-ID:
  Owner-ID     ===> TCHTST__
  Request Name ===> EXAMPLE8   (Required for options D, M, and S)


Enter Request-ID (if required) and make a selection, then press ENTER.
```

Press **PF3**

When the coworker leaves for the day, he/she will use the **Post Event** dialog to post the event as follows:

Access the **PJS Main Menu** panel from your **ISPF Primary Option Menu** panel, or by what ever other means your installation uses. Then select option **E** to access the **PJS Event Menu** panel.

```
 05/01/26 17:39 -------  PJS - PERSONAL JOB SCHEDULER  ---------------- 3.1.0
 OPTION ===> E_____________________________________________________________

Select one of the following functions:

  J  JOBREQ   - Update Job Requests
  C  CALENDAR - Update Calendars
  E  EVENT    - Update Events
  V  VARIABLE - Update Variables

  T  TUTORIAL - Enter PJS/ISPF Tutorial


Select an option, then press ENTER.


(c) Copyright, Northrop Grumman, 1990, 2005.
For additional copyright information enter ABOUT on the command line.

PJS is distributed under the GNU General Public License.
For license information enter LICENSE on the command line.
```

Press **ENTER**

From the **Event Menu** panel, select option **L** to access the **List Events** panel. The **Owner-ID,** defaults to the TSO User-ID (which is now USER02). The **Event Name** is not used.

```
05/01/26 17:39 -------  PJS - EVENT MENU  -------------------------------------
OPTION ===> L

Select one of the following functions:

  L  LIST     - List all Events for Owner-ID
  P  POST     - Post an Event
  R  RESET    - Reset an Event
  S  DISPLAY  - Display an Event


Event-ID:
  Owner-ID      ===> USER02__
  Event Name    ===> ________       (Required for options A, D, M, and S)


Enter Event-ID (if required) and make a selection, then press ENTER.
```

Press **ENTER**

On the **List Events** panel, enter a **P** line command to access the **Post Event** panel for the event.

```
05/01/26 17:39 -------  PJS - LIST EVENTS  ----------------- LINE 1    OF 1
COMMAND ===> ____________________________________________________ SCROLL ===> PAGE

S     Event-ID       Last Post Date/Time     Status
P USER02.IMGONE

Line Commands ==> P - Post Event, R - Reset Event, S - Display Event
                  J - List Job Requests for Event
```

Press **ENTER**

On the **Post Event** panel, enter **YES** in the confirmation field to confirm the post, then press **PF3** to complete the post.

```
05/01/26 17:39 -------  PJS - POST EVENT  -------------------------------------
COMMAND ===> ____________________________________________________________________

Event-ID         ===> USER02.IMGONE

Event Status     ===>

Date/Time Posted ===>


Enior 'YES' and press enter to confirm post ===> YES









Commands ==> REQ - Display Job Requests for Event,
             END - Complete Post, CANCEL - Cancel Post
```

Press **PF3**

The event post is now complete. The event status is POST PENDING. It will remain in this status for a short period of time until the PJS System Task posts all the job request events that specify this event. If it is after 5:00 PM, the JCL for the job request above will be submitted.

```
05/01/26 17:39 -------  PJS - LIST EVENTS  ---------------------- Event posted
COMMAND ===> ____________________________________________________ SCROLL ===> PAGE

S     Event-ID       Last Post Date/Time     Status
_ USER02.IMGONE                            POST PENDING









Line Commands ==> P - Post Event, R - Reset Event, S - Display Event
                  J - List Job Requests for Event
```

Press **PF3**

Example 9: How to Enable a Failed Job Request

Problem

In this example, suppose the PJS System task encountered an error when processing one of your job requests. When this happens, a message describing the error is sent to the user, and the job request is placed in the ERROR status. After fixing the error, you must '**enable**' the job request before PJS can process it again.

When a job request is placed in the ERROR status, the **Next Run Date and Time** is not recalculated, and the **Job Request Events** are not reset. If the job request is enabled without recalculating the Next Run Date and Time, or without resetting the Job Request Events, the job will be ready to be submitted by PJS immediately. In some cases, this may be desirable. In others, it may not. In this example, these will be reset.

TSO Solution

Use the **PJREQLST** command to check the status of the job request as follows:

```
 READY
pjreqlst id(example9)
    Request-ID          Description
 TCHTST.EXAMPLE9    Example 9 - Enabling a Failed Job Request

                        Next Run        Window        Notify    Notify
           Status      Date    Time   Time  Option    Userid    Level
          ERROR     01/25/2005 17:00                 TCHTST    INFO

                   JCL Data Set Name                                Save
          TCHTST.MISC.JCL(MYJOB)                                   NOSAVE

             Last Submit          Status Change      Spool      Spool    Retry
            Date      Time       Date      Time      Member   Rec Count  Count
          01/25/2005 17:00:50  01/25/2005 17:00:50                    0      0

          Userid   Group
          TCHTST   WTST

                                     Start              End
             Frequency             Date    Time      Date    Time
            1 DAYS               01/25/2005 17:00

                             Pre-        Posted
             Event-ID        Post    Date      Time
          TCHTST.EVENT1      YES  01/25/2005 16:47:38

 READY
```

Use the **PJREQHST** command to check the messages that were issued for the job request as follows:

```
 READY
pjreqhst example9
    Request-ID          Description
 TCHTST.EXAMPLE9    Example 9 - Enabling a Failed Job Request

    Date     Time      Msg ID        Message Text
 01/25/2005 17:00:54  PJS4301E  JCL member not found, dsn=TCHTST.MISC.JCL(MYJOB)
 01/25/2005 17:00:55  PJS1502E  Job submit for request TCHTST.EXAMPLE9 failed
 01/25/2005 17:00:56  PJS1509I  Job request TCHTST.EXAMPLE9 has been disabled
                                in error
 READY
```

In this case the messages indicate that the JCL member was not found in the indicated data set. The JCL has been either renamed or deleted. To correct the error, restore the JCL, then enable the PJS job request.

Use the **PJREQMOD** command to update the job request, recalculating the Next Run date and Time, resetting the Job Request Events, and enabling the job request as follows:

```
 READY
pjreqmod example9
 REQ MOD
set enabled
 REQ MOD
reset runtime
 REQ MOD
event reset event1
 REQ MOD
list
    Request-ID          Description
 TCHTST.EXAMPLE9    Example 9 - Enabling a Failed Job Request

                        Next Run        Window        Notify    Notify
            Status      Date    Time   Time  Option    Userid    Level
           WAIT      01/26/2005 17:00                TCHTST    INFO

                    JCL Data Set Name                                Save
           TCHTST.MISC.JCL(MYJOB)                                   NOSAVE

              Last Submit          Status Change      Spool      Spool    Retry
             Date      Time       Date      Time      Member   Rec Count  Count
           01/25/2005 17:00:50        PENDING                           0      0

           Userid   Group
           TCHTST   WTST

                                     Start              End
               Frequency             Date    Time      Date    Time
             1 DAYS               01/25/2005 17:00

                            Pre-        Posted
               Event-ID     Post    Date      Time
           TCHTST.EVENT1    YES

 REQ MOD
end
 PJS2102I  Job request TCHTST.EXAMPLE9 successfully modified
 READY
```

ISPF Solution

Access the **PJS Main Menu** panel from your **ISPF Primary Option Menu** panel, or by what ever other means your installation uses. Then select option **J** to access the **PJS Job Request Menu** panel.

```
 05/01/26 10:49 -------  PJS - PERSONAL JOB SCHEDULER  ---------------- 3.1.0
 OPTION ===> J_____________________________________________________________________

Select one of the following functions:

  J  JOBREQ   - Update Job Requests
  C  CALENDAR - Update Calendars
  E  EVENT    - Update Events
  V  VARIABLE - Update Variables

  T  TUTORIAL - Enter PJS/ISPF Tutorial


Select an option, then press ENTER.


(c) Copyright, Northrop Grumman, 1990, 2005.
For additional copyright information enter ABOUT on the command line.

PJS is distributed under the GNU General Public License.
For license information enter LICENSE on the command line.
```

Press **ENTER**

From the **Job Request Menu** panel, select option **L** to access the **List Job Requests** panel. The **Owner-ID,** defaults to the TSO User-ID. The **Request Name** is not used.

```
05/01/26 10:49 -------  PJS - JOB REQUEST MENU  -------------------------------
OPTION ===> L____________________________________________________________________

Select one of the following functions:

  L  LIST     - List all Job Requests for Owner
  A  ADD      - Add a new Job Request
  D  DELETE   - Delete a Job Request
  M  MODIFY   - Modify a Job Request
  S  DISPLAY  - Display a Job Request


Request-ID:
  Owner-ID     ===> TCHTST__
  Request Name ===> ________   (Required for options D, M, and S)


Enter Request-ID (if required) and make a selection, then press ENTER.
```

Press **ENTER**

The **List Job Requests** panel includes the status. In this case, we can see that one of our job requests (TCHTST.EXAMPLE9) is in the **ERROR** status.

On the **List Job Requests** panel, enter an **S** line command to access the **Display Job Request** panel for the job request.

```
05/01/26 10:49 -------  PJS - LIST JOB REQUESTS  ----------- LINE 1    OF 17
COMMAND ===> ____________________________________________________________________

S    Request-ID          Description                                    Status
_ TCHTST.DAILY      Example 2 - How to Run a Daily Job                 WAIT
_ TCHTST.EXAMPLE3   Example 3 - How to Run a Job Using a Calendar      WAIT
_ TCHTST.EXAMPLE6   Example 6 - Run a Job After CICS Terminates        WAIT
_ TCHTST.EXAMPLE8   Example 8 - Manually Posting an Event              COMPLETE
S TCHTST.EXAMPLE9   Example 9 - Enabling a Failed Job Request          ERROR
_ TCHTST.EXAMPL10   Example 10 - Modifying the Submitted JCL           WAIT
_ TCHTST.EXAMPL11   Example 11 - How to Use a Global Variable          WAIT
_ TCHTST.EX4JOB1    Example 4 - Job 1 - Using Events                   WAIT
_ TCHTST.EX4JOB2    Example 4 - Job 2 - Using Events                   WAIT
_ TCHTST.EX5JOB1    Example 5 - Job 1 - Using Advanced Events          WAIT
_ TCHTST.EX5JOB2    Example 5 - Job 2 - Using Advanced Events          WAIT
_ TCHTST.EX5JOB3    Example 5 - Job 3 - Using Advanced Events          WAIT
_ TCHTST.EX5JOB4    Example 5 - Job 4 - Using Advanced Events          WAIT
_ TCHTST.EX5JOB5    Example 5 - Job 5 - Using Advanced Events          WAIT

Commands ==> ADD - Add a new request
Line Commands ==> C - Copy Request, D - Delete Request, M - Modify Request,
                  S - Display Request
```

Press **ENTER**

From the **Display Job Request** panel, enter the **HIST** command to access the **Display Job Request History** panel.

```
05/01/26 10:49 -------  PJS - DISPLAY JOB REQUEST  ----------------------------
COMMAND ===> HIST_____________________________________________________________

Request-ID   ===> TCHTST.EXAMPLE9      Status       ===> ERROR
Description  ===> Example 9 - Enabling a Failed Job Request

JCL Data Set ===> TCHTST.MISC.JCL(MYJOB)

Next Run D/T ===> 01/25/2005 17:00     Last Run D/T ===> 01/25/2005 17:00:50
Window       ===>

Start D/T    ===> 01/25/2005 17:00     End D/T      ===>

Frequency    ===> 001 DAYS

Events       ===> POSTED



Commands ==> EVENT - Events, VAR - Variables, HIST - Request History,
             JCL - Browse JCL
```

Press **ENTER**

The **Display Job Request History** panel displays the messages issued by the PJS System Task for the job request. Be aware that if the message text exceeds the width of your screen you can see the rest of the message by scrolling left and right (usually with the **PF10** and **PF11** keys). These messages should indicate the cause of the error. In this case the messages indicate that the JCL member was not found in the indicated data set. The JCL has been either renamed or deleted.

After fixing the cause of the error, return to the Display Job Request panel by entering the **END** command, then continue back to the List Job Requests for Owner panel by entering the **END** command again (we will use **PF3** in both cases).

```
 05/01/26 10:49 -------  PJS - JOB REQUEST HISTORY  --------- LINE 1    OF 3
 COMMAND ===>                                                 SCROLL ===> PAGE

 Request-ID   ===> TCHTST.EXAMPLE9      Status       ===> ERROR
 Description  ===> Example 9 - Enabling a Failed Job Request

    Date      Time      Msg ID        Message Text
01/25/2005 17:00:54  PJS4301E  JCL member not found, dsn=TCH03.MISC.JCL(MYJOB)
01/25/2005 17:00:55  PJS1502E  Job submit for request TCH03.EXAMPLE9 failed
01/25/2005 17:00:56  PJS1509I  Job request TCH03.EXAMPLE9 has been disabled in









 Commands ==>  EVENT - Events, VAR - Variables, JCL - Browse JCL
```

Press **PF3** twice

On the **List Job Requests** panel, enter an **M** line command to access the **Modify Job Request** panel for the job request.

```
05/01/26 11:08 -------  PJS - LIST JOB REQUESTS  ----------- LINE 1    OF 17
COMMAND ===> ____________________________________________________________________

S    Request-ID          Description                                    Status
_ TCHTST.DAILY       Example 2 - How to Run a Daily Job                WAIT
_ TCHTST.EXAMPLE3    Example 3 - How to Run a Job Using a Calendar     WAIT
_ TCHTST.EXAMPLE6    Example 6 - Run a Job After CICS Terminates       WAIT
_ TCHTST.EXAMPLE8    Example 8 - Manually Posting an Event             COMPLETE
M TCHTST.EXAMPLE9    Example 9 - Enabling a Failed Job Request         ERROR
_ TCHTST.EXAMPL10    Example 10 - Modifying the Submitted JCL          WAIT
_ TCHTST.EXAMPL11    Example 11 - How to Use a Global Variable         WAIT
_ TCHTST.EX4JOB1     Example 4 - Job 1 - Using Events                  WAIT
_ TCHTST.EX4JOB2     Example 4 - Job 2 - Using Events                  WAIT
_ TCHTST.EX5JOB1     Example 5 - Job 1 - Using Advanced Events         WAIT
_ TCHTST.EX5JOB2     Example 5 - Job 2 - Using Advanced Events         WAIT
_ TCHTST.EX5JOB3     Example 5 - Job 3 - Using Advanced Events         WAIT
_ TCHTST.EX5JOB4     Example 5 - Job 4 - Using Advanced Events         WAIT
_ TCHTST.EX5JOB5     Example 5 - Job 5 - Using Advanced Events         WAIT

Commands ==> ADD - Add a new request
Line Commands ==> C - Copy Request, D - Delete Request, M - Modify Request,
                  S - Display Request
```

Press ENTER

On the **Modify Job Request** panel, enter **YES** in the **Enabled** field, then enter the **FREQ** command to access the **Specify Job Request Frequency** panel.

```
05/01/26 11:08 -------  PJS - MODIFY JOB REQUEST  ----------------------------
COMMAND ===> FREQ_____________________________________________________________

Request-ID   ===> TCHTST.EXAMPLE9
Description  ===> Example 9 - Enabling a Failed Job Request________

Status       ===> ERROR                Enabled      ===> YES  (Yes or No)

JCL Data Set ===> TCHTST.MISC.JCL(JOB1)

Next Run D/T ===> 01/25/2005 17:00     Last Run D/T ===> 01/25/2005 17:00:50
Window       ===>

Start D/T    ===> 01/25/2005 17:00     End D/T      ===>

Frequency    ===> 001 DAYS

Events       ===> POSTED

Commands ==> JCL - JCL Source, FREQ - Frequency, EVENT - Events,
             OPT - Submit Options, VAR - Variables, HIST - Request History,
             END - Complete Modify, CANCEL - Cancel Modify
```

Press **ENTER**

On the **Specify Job Request Frequency** panel, enter the YES in the Recalculate field. If this is not done, the Next Run date and Time will still indicate yesterday at 5:00 PM. Since this time is already past the job will be submitted as soon as we complete the update.

```
05/01/26 11:08 -------  PJS - SPECIFY JOB FREQUENCY  -------------------------
COMMAND ===> ______________________________________________________________________

Request-ID   ===> TCHTST.EXAMPLE9      Status       ===> WAIT
Description  ===> Example 9 - Enabling a Failed Job Request

Next Run D/T ===> 01/25/2005 17:00     Recalculate  ===> YES  (Yes or No)

Start D/T    ===> 01/25/2005 17:00     End D/T      ===> __________ ________

Frequency (choose one of the options below)
   Periodic     ===> 001 (num) DAYS___ (units - Yr, Mo, Wk, Day, Hr, or Min)
      or
   Day of Week  ===> _ Sun  _ Mon  _ Tue  _ Wed  _ Thu  s Fri  _ Sat
      or
   End of Month ===> Last Day - __   (days before last day of each month)
      or
   Calendar(s)  ===> _________________ _________________ _________________
      or
   Once (if none of the above is specified)

Commands ==> JCL - JCL Source, EVENT - Events, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
```

Press **ENTER**

After pressing **ENTER**, the **Next Run Date and Time** is recalculated. Enter the **EVENT** command to access the **Specify Job Request Events** panel.

```
05/01/26 11:08 -------  PJS - SPECIFY JOB FREQUENCY  --------------------------
COMMAND ===> EVENT_____________________________________________________________

Request-ID   ===> TCHTST.EXAMPLE9      Status       ===> WAIT
Description  ===> Example 9 - Enabling a Failed Job Request

Next Run D/T ===> 01/26/2005 17:00     Recalculate  ===> ___  (Yes or No)

Start D/T    ===> 01/25/2005 17:00     End D/T      ===> _________ ________

Frequency (choose one of the options below)
   Periodic     ===> 001 (num) DAYS___ (units - Yr, Mo, Wk, Day, Hr, or Min)
      or
   Day of Week  ===> _ Sun  _ Mon  _ Tue  _ Wed  _ Thu  s Fri  _ Sat
      or
   End of Month ===> Last Day - __   (days before last day of each month)
      or
   Calendar(s)  ===> _________________ _________________ _________________
      or
   Once (if none of the above is specified)

Commands ==> JCL - JCL Source, EVENT - Events, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
```

Press **ENTER**

On the **Specify Job Request Events** panel, the **Date/Time Posted** indicates that the job request event is posted. Enter an **R** line command to reset the job request event.

```
05/01/26 11:08 ------  PJS - SPECIFY JOB REQUEST EVENTS  --- LINE 1    OF 1
COMMAND ===> ____________________________________________________ SCROLL ===> PAGE

Request-ID   ===> TCHTST.EXAMPLE9      Status       ===> WAIT
Description  ===> Example 9 - Enabling a Failed Job Request

S      Event-ID     Prepost  Date/Time Posted
R  TCHTST.EVENT1      YES    01/25/2005 16:47:38
_  ________________   ___
_  ________________   ___
_  ________________   ___
_  ________________   ___
_  ________________   ___
_  ________________   ___
_  ________________   ___
_  ________________   ___
_  ________________   ___

Commands ==> JCL - JCL Source, FREQ - Frequency, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
Line Commands ==> D - Delete Event, P - Post Job Event, R - Reset Job Event
```

Press **ENTER**

After pressing **ENTER**, the job request event is reset, and the Date/Time Posted is cleared. Return to the **Modify Job Request** panel by entering the **END** command (we will use **PF3**).

```
05/01/26 11:08 ------  PJS - SPECIFY JOB REQUEST EVENTS  --- LINE 1    OF 1
COMMAND ===> ____________________________________________________ SCROLL ===> PAGE

Request-ID   ===> TCHTST.EXAMPLE9      Status       ===> WAIT
Description  ===> Example 9 - Enabling a Failed Job Request

S      Event-ID     Prepost  Date/Time Posted
_   TCHTST.EVENT1      YES
_   ________________   ___
_   ________________   ___
_   ________________   ___
_   ________________   ___
_   ________________   ___
_   ________________   ___
_   ________________   ___
_   ________________   ___
_   ________________   ___

Commands ==> JCL - JCL Source, FREQ - Frequency, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
Line Commands ==> D - Delete Event, P - Post Job Event, R - Reset Job Event
```

Press **PF3**

We have now finished the clean-up for this job request. We now complete the modify and return to the After fixing the cause of the error, return to the List Job Requests for Owner panel by entering the **END** command (once again, we use **PF3**).

```
05/01/26 11:08 -------  PJS - MODIFY JOB REQUEST  ----------------------------
COMMAND ===> ____________________________________________________________________

Request-ID   ===> TCHTST.EXAMPLE9
Description  ===> Example 9 - Enabling a Failed Job Request________

Status       ===> WAIT                 Enabled      ===> YES  (Yes or No)

JCL Data Set ===> TCHTST.MISC.JCL(JOB1)

Next Run D/T ===> 01/26/2005 17:00     Last Run D/T ===> 01/25/2005 17:00:50
Window       ===>

Start D/T    ===> 01/25/2005 17:00     End D/T      ===>

Frequency    ===> 001 DAYS

Events       ===> 00 OF 01

Commands ==> JCL - JCL Source, FREQ - Frequency, EVENT - Events,
             OPT - Submit Options, VAR - Variables, HIST - Request History,
             END - Complete Modify, CANCEL - Cancel Modify
```

Press **PF3**

The job request has now be reset. PJS will attempt to submit the job at its next scheduled run date and time.

```
05/01/26 11:08 -------  PJS - LIST JOB REQUESTS  --------- Job request modified
COMMAND ===> ____________________________________________________________________

S    Request-ID          Description                                      Status
_ TCHTST.DAILY      Example 2 - How to Run a Daily Job                  WAIT
_ TCHTST.EXAMPLE3   Example 3 - How to Run a Job Using a Calendar       WAIT
_ TCHTST.EXAMPLE6   Example 6 - Run a Job After CICS Terminates         WAIT
_ TCHTST.EXAMPLE8   Example 8 - Manually Posting an Event               COMPLETE
_ TCHTST.EXAMPLE9   Example 9 - Enabling a Failed Job Request           WAIT
_ TCHTST.EXAMPL10   Example 10 - Modifying the Submitted JCL            WAIT
_ TCHTST.EXAMPL11   Example 11 - How to Use a Global Variable           WAIT
_ TCHTST.EX4JOB1    Example 4 - Job 1 - Using Events                    WAIT
_ TCHTST.EX4JOB2    Example 4 - Job 2 - Using Events                    WAIT
_ TCHTST.EX5JOB1    Example 5 - Job 1 - Using Advanced Events           WAIT
_ TCHTST.EX5JOB2    Example 5 - Job 2 - Using Advanced Events           WAIT
_ TCHTST.EX5JOB3    Example 5 - Job 3 - Using Advanced Events           WAIT
_ TCHTST.EX5JOB4    Example 5 - Job 4 - Using Advanced Events           WAIT
_ TCHTST.EX5JOB5    Example 5 - Job 5 - Using Advanced Events           WAIT

Commands ==> ADD - Add a new request
Line Commands ==> C - Copy Request, D - Delete Request, M - Modify Request,
                  S - Display Request
```

Press PF3 twice

Example 10: How to Modify the Submitted JCL at Submit Time

Problem

In this example, suppose you want to submit a job to remind you of a weekly staff meeting, held every Thursday at 1:30 PM. Also, suppose that you wish to use the same JCL to remind you of other events that may occur throughout the day.

Before defining the job requests to PJS, first place the JCL for the SEND job in your JCL data set. Our JCL will look like this:

```
//TSOMSG   JOB account,TSOMSG
//*
//STEP1   EXEC PGM=IKJEFT01,DYNAMNBR=25
//SYSTSPRT DD  SYSOUT=*
//SYSTSIN  DD  *
 SEND -
 '$DATE $TIME $MSG' -
 USER($USER) $OPT
/*
```

The following substitutions are to be made:

$DATE is to be replaced with the current date.

$TIME is to be replaced with the current time.

$MSG is to be replaced with the message text.

$USER is to be replaced with the TSO User-ID to which the message is to be sent.

$OPT is to be replaced with the send command options (e.g. **LOGON**).

TSO Solution

Use the **PJREQADD** command to add the job request as follows:

```
 READY
pjreqadd id(exampl10)
 REQ ADD
set desc('Example 10 - Modifying the Submitted JCL')
 REQ ADD
set jcldsn('tchtst.misc.jcl(tsomsg)')
 REQ ADD
set weekday(thurs) strtdate(*) strttime(1:30PM)
 REQ ADD
variable add target('$DATE') dyn(sysdate) shift(all)
 REQ ADD
variable add target('$TIME') dyn(systime) shift(all)
 REQ ADD
variable add target('$MSG') lit('Weekly Staff Meeting') shift(all)
 REQ ADD
variable add target('$USER') lit('TCHTST') shift(all)
 REQ ADD
variable add target('$OPT') lit('NOW') shift(all)
 REQ ADD
end
 PJS2101I  Job request TCHTST.EXAMPL10 successfully added
 READY
pjreqlst id(exampl10)
    Request-ID          Description
 TCHTST.EXAMPL10    Example 10 - Modifying the Submitted JCL

                         Next Run        Window        Notify    Notify
            Status      Date    Time   Time  Option    Userid    Level
           WAIT      01/27/2005 13:30                 TCHTST    INFO

                    JCL Data Set Name                                Save
           TCHTST.MISC.JCL(TSOMSG)                                  NOSAVE

              Last Submit          Status Change      Spool      Spool    Retry
             Date      Time       Date      Time      Member   Rec Count  Count
                                01/25/2005 15:50:48                    0      0

           Userid   Group
           TCHTST   WTST

                                     Start              End
               Frequency            Date    Time      Date    Time
           TH                    01/25/2005 13:30

            Target   Search   Shift    Repl      Replacement Value
            String  Strt End Type Col  Type      Text/Variable/Dynamic
           $DATE      1  80  ALL   80  DYN  SYSDATE
           $TIME      1  80  ALL   80  DYN  SYSTIME
           $MSG       1  80  ALL   80  LIT  Weekly Staff Meeting
           $USER      1  80  ALL   80  LIT  TCHTST
           $OPT       1  80  ALL   80  LIT  NOW

 READY
```

ISPF Solution

Use the **Add Job Request** dialog to add the job request as follows:

Access the **PJS Main Menu** panel from your **ISPF Primary Option Menu** panel, or by what ever other means your installation uses. Then select option **J** to access the **PJS Job Request Menu** panel.

```
 05/01/26 13:14 -------  PJS - PERSONAL JOB SCHEDULER  ---------------- 3.1.0
 OPTION ===> J_____________________________________________________________

Select one of the following functions:

  J  JOBREQ   - Update Job Requests
  C  CALENDAR - Update Calendars
  E  EVENT    - Update Events
  V  VARIABLE - Update Variables

  T  TUTORIAL - Enter PJS/ISPF Tutorial


Select an option, then press ENTER.


(c) Copyright, Northrop Grumman, 1990, 2005.
For additional copyright information enter ABOUT on the command line.

PJS is distributed under the GNU General Public License.
For license information enter LICENSE on the command line.
```

Press **ENTER**

On the **PJS Job Request Menu** panel, enter the **Request Name**, then select option **A** to access the **Add Job Request** panel. The **Owner-ID** defaults to the TSO User-ID.

```
05/01/26 13:14 -------  PJS - JOB REQUEST MENU  -------------------------------
OPTION ===> A____________________________________________________________________

Select one of the following functions:

  L  LIST     - List all Job Requests for Owner
  A  ADD      - Add a new Job Request
  D  DELETE   - Delete a Job Request
  M  MODIFY   - Modify a Job Request
  S  DISPLAY  - Display a Job Request


Request-ID:
  Owner-ID     ===> TCHTST__
  Request Name ===> EXAMPL10   (Required for options D, M, and S)


Enter Request-ID (if required) and make a selection, then press ENTER.
```

Press **ENTER**

From the **Add Job Request** panel, enter the **Request Description**, then enter the **JCL** command to access the **Specify Job Request JCL Source** panel.

```
05/01/26 13:14 -------  PJS - ADD JOB REQUEST  --------------------------------
COMMAND ===> JCL_____________________________________________________________

Request-ID   ===> TCHTST.EXAMPL10
Description  ===> Example 10 - Modifying the Submitted JCL__________

Status       ===> WAIT                  Enabled      ===> YES  (Yes or No)

JCL Data Set ===>

Next Run D/T ===>
Window       ===>

Start D/T    ===>                       End D/T      ===>

Frequency    ===> ONCE

Events       ===> NONE

Commands ==> JCL - JCL Source, FREQ - Frequency, EVENT - Events,
             OPT - Submit Options, VAR - Variables,
             END - Complete Add, CANCEL - Cancel Add
```

Press **ENTER**

On the **Specify Job Request JCL Source** panel, enter the **JCL Data Set Name** and the **Member Name** in which the JCL has been placed. The **JCL Save Option** defaults to **NO**.

```
05/01/26 13:14 -------  PJS - SPECIFY JCL SOURCE  ----------------------------
COMMAND ===> ____________________________________________________________________

Request-ID   ===> TCHTST.EXAMPL10      Status       ===> WAIT
Description  ===> Example 10 - Modifying the Submitted JCL


JCL Data Set Name ===> TCHTST.MISC.JCL_____________________________
    Member Name   ===> TSOMSG__

JCL Save Option   ===> NO_____      (Yes, No, or Refresh)





Commands ==> BROWSE - Browse JCL, BRDATA - Browse JCL Data Set,
             FREQ - Frequency, EVENT - Events, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
```

Press **ENTER**

After the information on the **Specify Job Request JCL Source** panel has been successfully entered, enter the **FREQ** command to access the **Specify Job Request Frequency** panel.

```
05/01/26 13:14 -------  PJS - SPECIFY JCL SOURCE  ----------------------------
COMMAND ===> FREQ_____________________________________________________________

Request-ID   ===> TCHTST.EXAMPL10      Status       ===> WAIT
Description  ===> Example 10 - Modifying the Submitted JCL


JCL Data Set Name ===> TCHTST.MISC.JCL_____________________________
    Member Name   ===> TSOMSG__

JCL Save Option   ===> NO_____      (Yes, No, or Refresh)





Commands ==> BROWSE - Browse JCL, BRDATA - Browse JCL Data Set,
             FREQ - Frequency, EVENT - Events, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
```

Press **ENTER**

On the **Specify Job Request Frequency** panel, enter the **Start Date and Time**. In this case, the **Start Date** is entered as a **relative date**, to specify 'today'. The Start Time is entered as an **absolute time** to specify the specific time at which the job is to be run. The **Frequency** is entered as a **Day-of-Week** frequency by entering a non-blank character in the unprotected field immediately before the weekday to be selected.

```
05/01/26 13:14 -------  PJS - SPECIFY JOB FREQUENCY  -------------------------
COMMAND ===> ____________________________________________________________________

Request-ID   ===> TCHTST.EXAMPL10      Status       ===> WAIT
Description  ===> Example 10 - Modifying the Submitted JCL

Next Run D/T ===>                      Recalculate  ===> ___  (Yes or No)

Start D/T    ===> *________ 11:30 am   End D/T      ===> _________ ________

Frequency (choose one of the options below)
   Periodic     ===> ___ (num) _______ (units - Yr, Mo, Wk, Day, Hr, or Min)
      or
   Day of Week  ===> _ Sun  _ Mon  _ Tue  _ Wed  s Thu  _ Fri  _ Sat
      or
   End of Month ===> Last Day - __   (days before last day of each month)
      or
   Calendar(s)  ===> _________________ _________________ _________________
      or
   Once (if none of the above is specified)

Commands ==> JCL - JCL Source, EVENT - Events, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
```

Press **ENTER**

After pressing **ENTER**, the **Next Run Date and Time** is calculated. Also note that the **relative date** we entered for the **Start Date** has been translated to an **absolute date**.

After the information on the **Specify Job Request Frequency** panel has been successfully entered, enter the **VAR** command to access the **Specify Job Request Variables** panel.

```
05/01/26 13:14 -------  PJS - SPECIFY JOB FREQUENCY  --------------------------
COMMAND ===> VAR_____________________________________________________________

Request-ID   ===> TCHTST.EXAMPL10      Status       ===> WAIT
Description  ===> Example 10 - Modifying the Submitted JCL

Next Run D/T ===> 01/27/2005 13:30     Recalculate  ===> ___  (Yes or No)

Start D/T    ===> 01/26/2005 13:30     End D/T      ===> _________ ________

Frequency (choose one of the options below)
   Periodic     ===> ___ (num) _______ (units - Yr, Mo, Wk, Day, Hr, or Min)
      or
   Day of Week  ===> _ Sun  _ Mon  _ Tue  S Wed  _ Thu  _ Fri  _ Sat
      or
   End of Month ===> Last Day - __   (days before last day of each month)
      or
   Calendar(s)  ===> _________________ _________________ _________________
      or
   Once (if none of the above is specified)

Commands ==> JCL - JCL Source, EVENT - Events, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
```

Press **ENTER**

On the **Specify Job Request Variables** panel, enter the **Target String**, **Shift Type**, **Replacement Type**, and **Replacement Value** for each variable substitution to be performed.

```
05/01/26 13:14 -------  PJS - JOB REQUEST VARIABLES  ------- LINE 0    OF 0
COMMAND ===> ____________________________________________________ SCROLL ===> PAGE

Request-ID   ===> TCHTST.EXAMPL10      Status       ===> WAIT
Description  ===> Example 10 - Modifying the Submitted JCL

    Target   Search   Shift    Repl
S   String  Strt End Type Col  Type  Replacement Value      Col 1  to 42 of 82
_  $DATE___  __  __  all_  __  dyn_  sysdate_________________________________
_  $TIME___  __  __  all_  __  dyn_  systime________________________________
_  $MSG____  __  __  all_  __  lit_  'Weekly Staff Meeting'__________________
_  $USER___  __  __  all_  __  lit_  TCHTST__________________________________
_  $OPT____  __  __  all_  __  lit_  NOW____________________________________
_  ________  __  __  ____  __  ____  ______________________________________
_  ________  __  __  ____  __  ____  ______________________________________
_  ________  __  __  ____  __  ____  ______________________________________
_  ________  __  __  ____  __  ____  ______________________________________

Commands ==> JCL - JCL Source, FREQ - Frequency, EVENT - Events,
             OPT - Submit Options, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
Line Commands ==> I - Insert Variable, D - Delete Variable, R - Repeat Variable
```

Press **ENTER**

After pressing **ENTER**, the **Event-ID** is protected. After the information on the **Specify Job Request Events** panel has been successfully entered, return to the **Add Job Request** panel by entering the **END** command (we will use **PF3**).

```
05/01/26 13:15 -------  PJS - JOB REQUEST VARIABLES  ------- LINE 1    OF 5
COMMAND ===> ____________________________________________________ SCROLL ===> PAGE

Request-ID   ===> TCHTST.EXAMPL10      Status       ===> WAIT
Description  ===> Example 10 - Modifying the Submitted JCL

    Target   Search   Shift    Repl
S   String  Strt End Type Col  Type  Replacement Value      Col 1  to 42 of 82
_  $DATE___  01  80  ALL_  80  DYN_  SYSDATE_____________________________________
_  $TIME___  01  80  ALL_  80  DYN_  SYSTIME_____________________________________
_  $MSG____  01  80  ALL_  80  LIT_  'Weekly Staff Meeting'_____________________
_  $USER___  01  80  ALL_  80  LIT_  'TCHTST'___________________________________
_  $OPT____  01  80  ALL_  80  LIT_  'NOW'______________________________________
_  ________  __  __  ____  __  ____  ______________________________________________
_  ________  __  __  ____  __  ____  ______________________________________________
_  ________  __  __  ____  __  ____  ______________________________________________
_  ________  __  __  ____  __  ____  ______________________________________________

Commands ==> JCL - JCL Source, FREQ - Frequency, EVENT - Events,
             OPT - Submit Options, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
Line Commands ==> I - Insert Variable, D - Delete Variable, R - Repeat Variable
```

Press **PF3**

We have now entered all the parameters we need for this job request. The **Add Job Request** panel has been updated to show the parameters we have entered. We now complete the add and return to the **Job Request Menu** panel by entering the **END** command (once again, we use **PF3**).

```
05/01/26 13:15 -------  PJS - ADD JOB REQUEST  --------------------------------
COMMAND ===> ____________________________________________________________________

Request-ID   ===> TCHTST.EXAMPL10
Description  ===> Example 10 - Modifying the Submitted JCL__________

Status       ===> WAIT                 Enabled      ===> YES  (Yes or No)

JCL Data Set ===> TCHTST.MISC.JCL(TSOMSG)

Next Run D/T ===> 01/27/2005 13:30
Window       ===>

Start D/T    ===> 01/26/2005 13:30     End D/T      ===>

Frequency    ===> TH

Events       ===> NONE

Commands ==> JCL - JCL Source, FREQ - Frequency, EVENT - Events,
             OPT - Submit Options, VAR - Variables,
             END - Complete Add, CANCEL - Cancel Add
```

Press **PF3**

The job request add is now complete. PJS will modify the JCL and submit the job at the scheduled time.

```
05/01/26 16:12 -------  PJS - JOB REQUEST MENU  ------------ Job request added
OPTION ===> ____________________________________________________________________

Select one of the following functions:

  L  LIST     - List all Job Requests for Owner
  A  ADD      - Add a new Job Request
  D  DELETE   - Delete a Job Request
  M  MODIFY   - Modify a Job Request
  S  DISPLAY  - Display a Job Request


Request-ID:
  Owner-ID     ===> TCHTST__
  Request Name ===> EXAMPL10   (Required for options D, M, and S)


Enter Request-ID (if required) and make a selection, then press ENTER.
```

Press **PF3**

Example 11: How to Use a Global Variable

Problem

In this example, suppose you want to run several jobs using dirrefent job requests that use the same control parmater. Rather that updating each job's JCL, or updating each job request individually, you can use a **Global Variable** to insert the same parameter into each job.

To use a Global Variable, we will first define the variable, then use it in the job request variables for each job that is to use the parameter. The JCL is updated to include a target string where the parameter is to be inserted. When the value of the parameter changes, we need only update the global variable, which will automatically be included in each job.

TSO Solution

Use the **PJVARADD** command to add the global variable as follows:

```
 READY
pjvaradd acctend desc('Accounting Month End Date') lit('01/28/2005')
    Variable-ID          Description
 TCHTST.ACCTEND     Accounting Month End Date

           Repl       Replacement Value
           Type       Text/Variable/Dynamic
           LIT   01/28/2005

 PJS2401I  Variable record TCHTST.ACCTEND  successfully added
 READY
```

Next, use the **PJREQADD** command to add a job request as follows:

```
 READY
pjreqadd id(exampl11)
 REQ ADD
set desc('Example 11 - How to Use a Global Variable')
 REQ ADD
set jcldsn('tchtst.misc.jcl(job1)')
 REQ ADD
set eom strtdate(*) strttime(6:00pm)
 REQ ADD
variable add target('$DATE') var(acctend)
 REQ ADD
end
 PJS2101I  Job request TCHTST.EXAMPL11 successfully added
 READY
pjreqlst id(exampl11)
    Request-ID          Description
 TCHTST.EXAMPL11    Example 11 - How to Use a Global Variable

                        Next Run        Window        Notify    Notify
           Status      Date    Time   Time  Option    Userid    Level
          WAIT      01/31/2005 18:00                 TCHTST    INFO

                   JCL Data Set Name                                Save
          TCHTST.MISC.JCL(JOB1)                                    NOSAVE

             Last Submit          Status Change      Spool      Spool    Retry
            Date      Time       Date      Time      Member   Rec Count  Count
                                01/25/2005 16:01:42                    0      0

          Userid   Group
          TCHTST   WTST

                                         Start              End
              Frequency              Date    Time      Date    Time
          EOM- 0                 01/25/2005 18:00

           Target   Search   Shift    Repl      Replacement Value
           String  Strt End Type Col  Type      Text/Variable/Dynamic
          $DATE      1  80  ALL   80  VAR  TCHTST.ACCTEND

 READY
```

After the job runs successfully, and before the next time the job runs, you should update the global variable to indicate the next Accoubting Month End Date. Use the **PJVARADD** command to update the global variable as follows:

```
 READY
pjvarmod acctend lit('02/25/2005')
    Variable-ID          Description
 TCHTST.ACCTEND     Accounting Month End Date

           Repl       Replacement Value
           Type       Text/Variable/Dynamic
           LIT   02/25/2005

 PJS2402I  Variable record TCHTST.ACCTEND  successfully modified
 READY
```

ISPF Solution

Use the **Add Global Variable** dialog to add the global variable as follows:

Access the **PJS Main Menu** panel from your **ISPF Primary Option Menu** panel, or by what ever other means your installation uses. Then select option **V** to access the **PJS Global Variable Menu** panel.

```
 05/01/26 14:42 -------  PJS - PERSONAL JOB SCHEDULER  ---------------- 3.1.0
 OPTION ===> V_____________________________________________________________________

Select one of the following functions:

  J  JOBREQ   - Update Job Requests
  C  CALENDAR - Update Calendars
  E  EVENT    - Update Events
  V  VARIABLE - Update Variables

  T  TUTORIAL - Enter PJS/ISPF Tutorial


Select an option, then press ENTER.


(c) Copyright, Northrop Grumman, 1990, 2005.
For additional copyright information enter ABOUT on the command line.

PJS is distributed under the GNU General Public License.
For license information enter LICENSE on the command line.
```

Press **ENTER**

On the **PJS Global Variable Menu** panel, enter the **Variable Name**, then select option **A** to access the **Add Global Variable** panel. The **Owner-ID** defaults to the TSO User-ID.

```
05/01/26 14:42 -------  PJS - VARIABLE MENU  ----------------------------------
OPTION ===> A

Select one of the following functions:

  L  LIST     - List all Variables for Owner-ID
  A  ADD      - Add a new Variables
  D  DELETE   - Delete a Variable
  M  MODIFY   - Modify a Variable
  S  DISPLAY  - Display a Variable


Variable-ID:
  Owner-ID      ===> TCHTST
  Variable Name ===> ACCTEND       (Required for options A, D, M, and S)


Enter Variable-ID (if required) and make a selection, then press ENTER.
```

Press **ENTER**

On the **Add Global Variable** panel, enter the **Description**, then enter the **Replacement Value**. The **Value Type** defaults to **LITERAL**.

```
05/01/26 14:42 -------  PJS - ADD VARIABLE  -----------------------------------
COMMAND ===> ____________________________________________________________________

Variable-ID  ===> TCHTST.ACCTEND
Description  ===> Accounting Month End Date_____________________________

Value Type   ===> LITERAL_      (Literal, Variable, or Dynamic)

Value        ===> 01/28/2005_____________________________________________________
_________________________

Commands ==> REQ - Display Job Requests for Variable,
              END - Complete Add, CANCEL - Cancel Add
```

Press **ENTER**

We will now complete the add and return to the **Global Variable Menu** panel by entering the **END** command (we will use **PF3**).

```
05/01/26 14:42 -------  PJS - ADD VARIABLE  -----------------------------------
COMMAND ===> ____________________________________________________________________

Variable-ID  ===> TCHTST.ACCTEND
Description  ===> Accounting Month End Date_________________________

Value Type   ===> LITERAL_      (Literal, Variable, or Dynamic)

Value        ===> '01/28/2005'_____________________________________________
____________________

Commands ==> REQ - Display Job Requests for Variable,
             END - Complete Add, CANCEL - Cancel Add
```

Press **PF3**

The global variable add is now complete, and the global variable can be used in a job request. Return to the **PJS Main Menu** panel by entering the **END** command (once again, we use **PF3**).

```
05/01/26 14:43 -------  PJS - VARIABLE MENU  ------------------ Variable added
OPTION ===> ____________________________________________________________________

Select one of the following functions:

  L  LIST     - List all Variables for Owner-ID
  A  ADD      - Add a new Variables
  D  DELETE   - Delete a Variable
  M  MODIFY   - Modify a Variable
  S  DISPLAY  - Display a Variable


Variable-ID:
  Owner-ID      ===> TCHTST__
  Variable Name ===> ACCTEND_       (Required for options A, D, M, and S)


Enter Variable-ID (if required) and make a selection, then press ENTER.
```

Press **PF3**

Next, use the **Add Job Request** dialog to add the job request as follows:

From the **PJS Main Menu** panel, select option **J** to access the **PJS Job Request Menu** panel.

```
 05/01/26 14:43 -------  PJS - PERSONAL JOB SCHEDULER  ---------------- 3.1.0
 OPTION ===> J_____________________________________________________________

Select one of the following functions:

  J  JOBREQ   - Update Job Requests
  C  CALENDAR - Update Calendars
  E  EVENT    - Update Events
  V  VARIABLE - Update Variables

  T  TUTORIAL - Enter PJS/ISPF Tutorial


Select an option, then press ENTER.


(c) Copyright, Northrop Grumman, 1990, 2005.
For additional copyright information enter ABOUT on the command line.

PJS is distributed under the GNU General Public License.
For license information enter LICENSE on the command line.
```

Press **ENTER**

On the **PJS Job Request Menu** panel, enter the **Request Name**, then select option **A** to access the **Add Job Request** panel. The **Owner-ID** defaults to the TSO User-ID.

```
05/01/26 14:43 -------  PJS - JOB REQUEST MENU  -------------------------------
OPTION ===> A____________________________________________________________________

Select one of the following functions:

  L  LIST     - List all Job Requests for Owner
  A  ADD      - Add a new Job Request
  D  DELETE   - Delete a Job Request
  M  MODIFY   - Modify a Job Request
  S  DISPLAY  - Display a Job Request


Request-ID:
  Owner-ID     ===> TCHTST__
  Request Name ===> EXAMPL11   (Required for options D, M, and S)


Enter Request-ID (if required) and make a selection, then press ENTER.
```

Press **ENTER**

From the **Add Job Request** panel, enter the **Request Description**, then enter the **JCL** command to access the **Specify Job Request JCL Source** panel.

```
05/01/26 14:43 -------  PJS - ADD JOB REQUEST  --------------------------------
COMMAND ===> JCL_____________________________________________________________

Request-ID   ===> TCHTST.EXAMPL11
Description  ===> Example 11 - How to Use a Global Variable________

Status       ===> WAIT                 Enabled      ===> YES  (Yes or No)

JCL Data Set ===>

Next Run D/T ===>
Window       ===>

Start D/T    ===>                      End D/T      ===>

Frequency    ===> ONCE

Events       ===> NONE

Commands ==> JCL - JCL Source, FREQ - Frequency, EVENT - Events,
             OPT - Submit Options, VAR - Variables,
             END - Complete Add, CANCEL - Cancel Add
```

Press **ENTER**

On the **Specify Job Request JCL Source** panel, enter the **JCL Data Set Name** and the **Member Name** in which the JCL has been placed. The **JCL Save Option** defaults to **NO**.

```
05/01/26 14:44 -------  PJS - SPECIFY JCL SOURCE  ----------------------------
COMMAND ===> ____________________________________________________________________

Request-ID   ===> TCHTST.EXAMPL11      Status       ===> WAIT
Description  ===> Example 11 - How to Use a Global Variable


JCL Data Set Name ===> TCHTST.MISC.JCL_____________________________
    Member Name   ===> JOB1____

JCL Save Option   ===> NO_____      (Yes, No, or Refresh)





Commands ==> BROWSE - Browse JCL, BRDATA - Browse JCL Data Set,
             FREQ - Frequency, EVENT - Events, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
```

Press **ENTER**

After the information on the **Specify Job Request JCL Source** panel has been successfully entered, enter the **FREQ** command to access the **Specify Job Request Frequency** panel.

```
05/01/26 14:44 -------  PJS - SPECIFY JCL SOURCE  ----------------------------
COMMAND ===> FREQ_____________________________________________________________

Request-ID   ===> TCHTST.EXAMPL11      Status       ===> WAIT
Description  ===> Example 11 - How to Use a Global Variable


JCL Data Set Name ===> TCHTST.MISC.JCL_____________________________
    Member Name   ===> JOB1____

JCL Save Option   ===> NO_____      (Yes, No, or Refresh)





Commands ==> BROWSE - Browse JCL, BRDATA - Browse JCL Data Set,
             FREQ - Frequency, EVENT - Events, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
```

Press **ENTER**

On the **Specify Job Request Frequency** panel, enter the **Start Date and Time** and the **Frequency**. In this case, the **Start Date** is entered as a **relative date**, to specify 'today'. The Start Time is entered as an **absolute time** to specify the specific time at which the job is to be run. The **Frequency** is entered as an **End-of-Month** frequency by entering the number of days before the last day of the month on which the job is to be run (in this case, zero).

```
05/01/26 14:44 -------  PJS - SPECIFY JOB FREQUENCY  -------------------------
COMMAND ===> ____________________________________________________________________

Request-ID   ===> TCHTST.EXAMPL11      Status       ===> WAIT
Description  ===> Example 11 - How to Use a Global Variable

Next Run D/T ===>                      Recalculate  ===> ___  (Yes or No)

Start D/T    ===> *________ 06:00 pm  End D/T      ===> _________ ________

Frequency (choose one of the options below)
   Periodic     ===> ___ (num) _______ (units - Yr, Mo, Wk, Day, Hr, or Min)
      or
   Day of Week  ===> _ Sun  _ Mon  _ Tue  _ Wed  _ Thu  _ Fri  _ Sat
      or
   End of Month ===> Last Day - 0_   (days before last day of each month)
      or
   Calendar(s)  ===> _________________ _________________ _________________
      or
   Once (if none of the above is specified)

Commands ==> JCL - JCL Source, EVENT - Events, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
```

Press **ENTER**

After pressing **ENTER**, the **Next Run Date and Time** is calculated. Even though the start date was enter as 'today', the next run date is set to the last day of the month. Also note that the **relative date** we entered for the **Start Date** has been translated to an **absolute date**.

After the information on the **Specify Job Request Frequency** panel has been successfully entered, enter the **VAR** command to access the **Specify Job Request Variables** panel.

```
05/01/26 14:44 -------  PJS - SPECIFY JOB FREQUENCY  --------------------------
COMMAND ===> VAR_____________________________________________________________

Request-ID   ===> TCHTST.EXAMPL11      Status       ===> WAIT
Description  ===> Example 11 - How to Use a Global Variable

Next Run D/T ===> 01/31/2005 18:00     Recalculate  ===> ___  (Yes or No)

Start D/T    ===> 01/26/2005 18:00     End D/T      ===> _________ ________

Frequency (choose one of the options below)
   Periodic     ===> ___ (num) _______ (units - Yr, Mo, Wk, Day, Hr, or Min)
      or
   Day of Week  ===> _ Sun  _ Mon  _ Tue  _ Wed  _ Thu  _ Fri  _ Sat
      or
   End of Month ===> Last Day - 00   (days before last day of each month)
      or
   Calendar(s)  ===> _________________ _________________ _________________
      or
   Once (if none of the above is specified)

Commands ==> JCL - JCL Source, EVENT - Events, OPT - Submit Options,
             VAR - Variables, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
```

Press **PF3**

On the **Specify Job Request Variables** panel, enter the **Target String**, **Shift Type**, **Replacement Type**, and **Replacement Value** for each variable substitution to be performed.

```
05/01/26 14:44 -------  PJS - JOB REQUEST VARIABLES  ------- LINE 0    OF 0
COMMAND ===> ____________________________________________________ SCROLL ===> PAGE

Request-ID   ===> TCHTST.EXAMPL11      Status       ===> WAIT
Description  ===> Example 11 - How to Use a Global Variable

    Target   Search   Shift    Repl
S   String  Strt End Type Col  Type  Replacement Value      Col 1  to 42 of 82
_  $DATE___  __  __  all_  __  var_  acctend_____________________________
_  ________  __  __  ____  __  ____  ________________________________________
_  ________  __  __  ____  __  ____  ________________________________________
_  ________  __  __  ____  __  ____  ________________________________________
_  ________  __  __  ____  __  ____  ________________________________________
_  ________  __  __  ____  __  ____  ________________________________________
_  ________  __  __  ____  __  ____  ________________________________________
_  ________  __  __  ____  __  ____  ________________________________________
_  ________  __  __  ____  __  ____  ________________________________________

Commands ==> JCL - JCL Source, FREQ - Frequency, EVENT - Events,
             OPT - Submit Options, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
Line Commands ==> I - Insert Variable, D - Delete Variable, R - Repeat Variable
```

Press **ENTER**

After pressing **ENTER**, the **Event-ID** is protected. After the information on the **Specify Job Request Events** panel has been successfully entered, return to the **Add Job Request** panel by entering the **END** command (we will use **PF3**).

```
05/01/26 14:45 -------  PJS - JOB REQUEST VARIABLES  ------- LINE 1    OF 1
COMMAND ===> ____________________________________________________ SCROLL ===> PAGE

Request-ID   ===> TCHTST.EXAMPL11      Status       ===> WAIT
Description  ===> Example 11 - How to Use a Global Variable

    Target   Search   Shift    Repl
S   String  Strt End Type Col  Type  Replacement Value      Col 1  to 42 of 82
_  $DATE___  01  80  ALL_  80  VAR_  TCHTST.ACCTEND_____________________________
_  ________  __  __  ____  __  ____  ________________________________________
_  ________  __  __  ____  __  ____  ________________________________________
_  ________  __  __  ____  __  ____  ________________________________________
_  ________  __  __  ____  __  ____  ________________________________________
_  ________  __  __  ____  __  ____  ________________________________________
_  ________  __  __  ____  __  ____  ________________________________________
_  ________  __  __  ____  __  ____  ________________________________________
_  ________  __  __  ____  __  ____  ________________________________________

Commands ==> JCL - JCL Source, FREQ - Frequency, EVENT - Events,
             OPT - Submit Options, HIST - Request History,
             END - Complete Updates, CANCEL - Cancel Updates
Line Commands ==> I - Insert Variable, D - Delete Variable, R - Repeat Variable
```

Press **PF3**

We have now entered all the parameters we need for our job request. The **Add Job Request** panel has been updated to show the parameters we have entered. We now complete the add and return to the **Job Request Menu** panel by entering the **END** command (once again, we use **PF3**).

```
05/01/26 14:45 -------  PJS - ADD JOB REQUEST  --------------------------------
COMMAND ===> ____________________________________________________________________

Request-ID   ===> TCHTST.EXAMPLE11
Description  ===> Example 11 - How to Use a Global Variable________

Status       ===> WAIT                 Enabled      ===> YES  (Yes or No)

JCL Data Set ===> TCHTST.MISC.JCL(JOB1)

Next Run D/T ===> 01/31/2005 18:00
Window       ===>

Start D/T    ===> 01/26/2005 18:00     End D/T      ===>

Frequency    ===> EOM-00

Events       ===> NONE

Commands ==> JCL - JCL Source, FREQ - Frequency, EVENT - Events,
             OPT - Submit Options, VAR - Variables,
             END - Complete Add, CANCEL - Cancel Add
```

Press **PF3**

The job request add is now complete. PJSwill modify the JCL and submit the job at the scheduled time.

```
05/01/26 14:45 -------  PJS - JOB REQUEST MENU  ------------ Job request added
OPTION ===> ____________________________________________________________________

Select one of the following functions:

  L  LIST     - List all Job Requests for Owner
  A  ADD      - Add a new Job Request
  D  DELETE   - Delete a Job Request
  M  MODIFY   - Modify a Job Request
  S  DISPLAY  - Display a Job Request


Request-ID:
  Owner-ID     ===> TCHTST__
  Request Name ===> EXAMPL11   (Required for options D, M, and S)


Enter Request-ID (if required) and make a selection, then press ENTER.
```

Press **PF3**

After the job runs successfully, and before the next time the job runs, you should update the global variable to indicate the next Accoubting Month End Date. Use the **Modify Global Variable** dialog to update the global variable as follows:

Access the **PJS Main Menu** panel from your **ISPF Primary Option Menu** panel, or by what ever other means your installation uses. Then select option **V** to access the **PJS Global Variable Menu** panel.

```
 05/02/07 09:12 -------  PJS - PERSONAL JOB SCHEDULER  ---------------- 3.1.0
 OPTION ===> V_____________________________________________________________

Select one of the following functions:

  J  JOBREQ   - Update Job Requests
  C  CALENDAR - Update Calendars
  E  EVENT    - Update Events
  V  VARIABLE - Update Variables

  T  TUTORIAL - Enter PJS/ISPF Tutorial


Select an option, then press ENTER.


(c) Copyright, Northrop Grumman, 1990, 2005.
For additional copyright information enter ABOUT on the command line.

PJS is distributed under the GNU General Public License.
For license information enter LICENSE on the command line.
```

Press **ENTER**

On the **PJS Global Variable Menu** panel, enter the **Variable Name**, then select option **A** to access the **Modify Global Variable** panel. The **Owner-ID** defaults to the TSO User-ID.

```
05/02/07 09:12 -------  PJS - VARIABLE MENU  ----------------------------------
OPTION ===> M

Select one of the following functions:

  L  LIST     - List all Variables for Owner-ID
  A  ADD      - Add a new Variables
  D  DELETE   - Delete a Variable
  M  MODIFY   - Modify a Variable
  S  DISPLAY  - Display a Variable


Variable-ID:
  Owner-ID      ===> TCHTST
  Variable Name ===> ACCTEND       (Required for options A, D, M, and S)


Enter Variable-ID (if required) and make a selection, then press ENTER.
```

Press **ENTER**

On the **Modify Global Variable** panel, enter the new **Replacement Value**.

```
05/02/07 09:12 -------  PJS - MODIFY VARIABLE  --------------------------------
COMMAND ===> ____________________________________________________________________

Variable-ID  ===> TCHTST.ACCTEND
Description  ===> Accounting Month End Date_____________________

Value Type   ===> LITERAL_      (Literal, Variable, or Dynamic)

Value        ===> '02/25/2005'_____________________________________________
____________________

Commands ==> REQ - Display Job Requests for Variable,
             END - Complete Add, CANCEL - Cancel Add
```

Press **ENTER**

We will now complete the update and return to the **Global Variable Menu** panel by entering the **END** command (we will use **PF3**).

```
05/02/07 09:12 -------  PJS - MODIFY VARIABLE  --------------------------------
COMMAND ===> ____________________________________________________________________

Variable-ID  ===> TCHTST.ACCTEND
Description  ===> Accounting Month End Date_________________________


Value Type   ===> LITERAL_      (Literal, Variable, or Dynamic)

Value        ===> '02/25/2005'_____________________________________________________
_________________________________________________________________________________
```

Press **PF3**

The global variable add is now complete, and the global variable can be used in a job request. Return to the **PJS Main Menu** panel by entering the **END** command (once again, we use **PF3**).

```
05/02/07 09:12 -------  PJS - VARIABLE MENU  --------------- Variable modified
OPTION ===> ____________________________________________________________________

Select one of the following functions:

  L  LIST     - List all Variables for Owner-ID
  A  ADD      - Add a new Variables
  D  DELETE   - Delete a Variable
  M  MODIFY   - Modify a Variable
  S  DISPLAY  - Display a Variable


Variable-ID:
  Owner-ID      ===> TCHTST__
  Variable Name ===> ACCTEND_       (Required for options A, D, M, and S)


Enter Variable-ID (if required) and make a selection, then press ENTER.
```

Press **PF3**

Appendix A: Summary of Changes

This appendix lists only the functional changes that will be of interest to PJS users. Installation changes and problems fixed are listed in the ***PJS Installation Guide***.

Changes for PJS Release 3.1

- PJS can now modify the JCL for a job request as it is being submitted. The user can specify a string appearing in the JCL that is to be replaced, and the replacement value. A replacement value may be a literal string, a **'Global Variable'** (which specifies a replacement value outside of the job request that uses it), or a **'Dynamic Value'** (which is computed at job submit time).
- A **PJS Message History Log** is provided to record messages sent to the user by the **PJS System Task**.
- The **Job Request-ID** can be specified by the user, instead of being generated by PJS. The Request Number is replaced by an alphanumeric **Request Name**. If a request name is not specified by the user when a job request is added, a numeric **Request Number** will still be generated by PJS.
- A 50-character **Job Request Description** may be specified for a job request. This description is used for documentation purposes only, and does not affect PJS processing in any way.
- A **Notify Message Level** may be specified for a job request, to specify the minimum message severity level for messages sent to the user by the PJS System Task. Messages not sent can still be retrieved from the **PJS Message History Log**.
- A **Notify Userid** may be specified for a job request, to specify the TSO Userid to which the PJS System Task is to send messages about the job request. The default is the **Owner-ID** of the job request.
- The user can now explicitly specify that the **Next Run Date and Time** is to be recalculated, without changing the **Start Date and Time**, **End Date and Time**, or **Frequency**.
- The PJS ISPF job request dialog panels have been reorganized to make room for new options, and to display more complete information about a job request as it is being added/modified. Several new sub-dialog panels have been implemented.

The **Add Job Request** and **Modify Job Request** panels now display summary information about the job request, while most job request parameters are updated on sub-dialog panels.

- The PJS **Job Request Add** and **Job Request Modify** TSO commands have been changed to use a sub-command processing mode. The job request to be added or modified is created or read into storage when the **PJREQADD** or **PJREQMOD** command is entered. One or more sub-commands are then used to update the job request. When the job request is complete, the **END** sub-command is entered to complete the add or modify.
- When the **Next Run Date and Time** is recalculated by the ISPF interface, it is recalculated immediately, rather than waiting until the add of modify is completed.
- A **Display Job Request Internal Information** ISPF panel is provided to display internal information about a job request. This information can be valuable in debugging problems.
- The **Job Request List** TSO command DETAIL format has been changed to include the Added Status Change Date/Time, Spool Member, Spool Record Count, and Retry Count.
- When a **Calendar-ID** is entered on the **Specify Job Request Frequency** panel, the Calendar-ID will be checked for validity immediately, rather than waiting until the add or modify is completed.
- A 50-character **Calendar Description** may be specified for a calendar. This description is used for documentation purposes only, and does not affect PJS processing in any way.
- Security access to Job Requests, Calendars and Events can now be checked when they are entered by the suer, instead of waiting until the add or modify is completed.
- The PJS messages have been renumbered. PJS messages are now numbered **PJS***nnnn* where *nnnn* is a 4 digit number.

Changes for PJS Release 2.1.4

- Commands to display copyright and license information have been added to the ISPF menu.

Changes for PJS Release 2.1.3

- The default century for 2-digit years is changed from 19*xx* to 20*xx*.

Changes for PJS Release 2.1

- PJS can process dates in the *dd*/*mm*/*yyyy* format (commonly used in Europe), in addition to the *mm*/*dd*/*yyyy* format (commonly used in the U.S.). The format used is controlled by the new **DATEFMT** parameter of the **PJSOPT** macro.
- PJS ISPF commands may be abbreviated to 2 characters.
- A PJS Tutorial Index is provided.
- The name of the Batch Event Post Utility is changed from '**PJSEVENT**' to '**PJSPOST**'. The old name, '**PJSEVENT**', is still supported as an alias of '**PJSPOST**'.
- A new Batch Event Reset Utility (**PJSRESET**) is available. This utility complements the Batch Event Post Utility (**PJSPOST**).
- The Batch Event Post Utility (**PJSPOST**) and the Batch Event Reset Utility (**PJSRESET**) are described in the PJS ISPF Tutorial.

GNU General Public License

Version 2, June 1991

Copyright © 1989, 1991 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
Everyone is permitted to copy and distribute verbatim copies of
this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program

or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

> **a)** You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
>
> **b)** You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
>
> **c)** If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
one line to give the program's name and a brief idea of what it does.
Copyright (C)  yyyy  name of author

This program is free software; you can redistribute it and/or modify it
under the terms of the GNU General Public License as published by the
Free Software Foundation; either version 2 of the License, or (at your
option) any later version.

This program is distributed in the hope that it will be useful, but
WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
General Public License for more details.

You should have received a copy of the GNU General Public License along
with this program; if not, write to the Free Software Foundation, Inc.,
59 Temple Place, Suite 330, Boston, MA  02111-1307  USA
```

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type
'show w'.
This is free software, and you are welcome to redistribute it under
certain conditions; type 'show c' for details.
```

The hypothetical commands 'show w' and 'show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than 'show w' and 'show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.

signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

GNU Free Documentation License

Version 1.2, November 2002

Copyright (C) 2000,2001,2002 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
Everyone is permitted to copy and distribute verbatim copies of
this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front

cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright (c)  YEAR  YOUR NAME.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.2 or
any later version published by the Free Software Foundation; with no
Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.  A
copy of the license is included in the section entitled "GNU Free
Documentation License".
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

```
with the Invariant Sections being LIST THEIR TITLES, with the Front-
Cover Texts being LIST, and with the Back-Cover Texts being LIST.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

PJS User's Guide