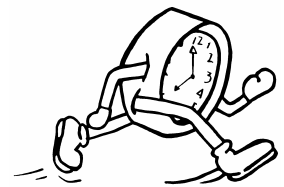


PJS^â

Personal Job Scheduler



User's Guide

Release 2.1

Related PJS Technical Documentation

Personal Job Scheduler (PJS®) Messages and Codes
Personal Job Scheduler (PJS®) Installation Guide

Personal Job Scheduler (PJS) 2.1.1 was released for distribution in April, 1992.
Personal Job Scheduler (PJS) 2.1.2 was released for distribution in November, 1992.
Personal Job Scheduler (PJS) 2.1.3 was released for distribution in October, 2002.
Personal Job Scheduler (PJS) 2.1.4 was released for distribution in February, 2004.

PJS is a registered trademark of Northrop Grumman.

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

Copyright © Northrop Grumman, 1990, 2004. 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

Preface	vii
Notational Symbols	viii
Acknowledgments	ix
PJS Software Copyright and License	ix
Comments and Suggestions	ix
 1. Introduction	 1
2. PJS System Overview	3
2.1 PJS Software Overview	3
2.1.1 The PJS Request Queue	3
2.1.2 The PJS JCL Spool	3
2.1.3 The PJS/TSO Interface	3
2.1.4 The PJS/ISPF Interface	4
2.1.5 The PJS System Task	4
2.2 Owners	4
2.3 PJS Job Requests	4
2.3.1 The request ID	4
2.3.2 Job Request Status	5
2.3.3 The JCL Data Set	5
2.3.4 Job Submission Date and Time	5
2.3.4.1 Start Date and Start Time	5
2.3.4.2 End Date and End Date	5
2.3.4.3 Next Run Date and Time	6
2.3.4.4 Last Run Date and Time	6
2.3.5 The Submit Window	6
2.3.6 Frequency	7
2.3.7 Job Request Events	7
2.4 PJS Calendars	8
2.5 PJS Events	8
2.5.1 The event ID	8
2.5.2 Events and Job Request Events	8
2.5.3 How to Post Events or Job Request Events	9
2.5.4 Preposted and Non Preposted Events	9
2.5.5 How to Reset Events or Job Request Events	10
2.6 PJS System Task Process Summary	11
 3. PJS Specification Conventions	 15
3.1 The owner ID	15
3.2 The request ID	16
3.3 The calendar ID	16
3.4 The event ID	16
3.5 Time Specifications	17
3.5.1 Absolute Time	17
3.5.2 Relative Time	17

3.6	Date Specifications	18
3.6.1	Absolute Date	18
3.6.2	Relative Date	18
3.7	Job Request Status Values	19
4.	PJS TSO Commands	21
4.1	PJS Job Request Commands	22
4.1.1	PJREQADD	23
4.1.1.1	How to Specify an OWNER	24
4.1.1.2	How to Specify the Data Set Name	24
4.1.1.3	How to Use the PJS JCL Spool	25
4.1.1.4	How to Specify the Date of Submission	25
4.1.1.5	How to Specify the Time of Submission	26
4.1.1.6	How to Specify a Frequency Option	26
4.1.1.7	How to Specify an End Date	29
4.1.1.8	How to Specify an End Time	29
4.1.1.9	How to Specify Job Request Events	29
4.1.1.10	How to Post Job Request Events	30
4.1.1.11	How to Specify a Submit Window	31
4.1.1.12	How to Disable the Job Request	32
4.1.2	PJREQDEL	33
4.1.2.1	How to Specify the request ID	33
4.1.3	PJREQLIST	34
4.1.3.1	How to List One Job Request	34
4.1.3.2	How to List All Job Requests For an Owner	34
4.1.3.3	How to List All Job Requests	35
4.1.3.4	How to Specify the Amount of Displayed Information	35
4.1.4	PJREQMOD	36
4.1.4.1	How to Specify the request ID	37
4.1.4.2	How to Modify the Data Set Name	37
4.1.4.3	How to Use the PJS JCL Spool	38
4.1.4.4	How to Specify a New Date of Submission	38
4.1.4.5	How to Specify a New Time of Submission	38
4.1.4.6	How to Specify a New Frequency Option	39
4.1.4.7	How to Specify, Replace, or Delete an End Date	42
4.1.4.8	How to Specify, Replace, or Delete an End Time	43
4.1.4.9	How to Specify, Add, or Delete Job Request Events	43
4.1.4.10	How to Post Job Request Events	45
4.1.4.11	How to Reset Job Request Events	45
4.1.4.12	How to Specify, Replace, or Delete a Submit Window	46
4.1.4.13	How to Disable or Enable the Job Request	47
4.2	PJS Calendar Commands	48
4.2.1	PJCALADD	49
4.2.1.1	How to Specify the calendar ID	49
4.2.1.2	How to Select Dates by Inclusion	49
4.2.1.3	How to Select Dates by Exclusion	50
4.2.2	PJCALDEL	51
4.2.2.1	How to Specify the calendar ID	51

4.2.3	PJCALIST	52
4.2.3.1	How to List One Calendar	52
4.2.3.2	How to List All Calendars for an Owner	52
4.2.3.3	How to List All Calendars	53
4.2.3.4	How to Specify the Amount of Displayed Information	53
4.2.4	PJCALMOD	54
4.2.4.1	How to Specify the calendar ID	54
4.2.4.2	How to Replace All Selected Dates by Inclusion	54
4.2.4.3	How to Replace All Selected Dates by Exclusion	55
4.2.4.4	How to Add Selected Dates to a Calendar	55
4.2.4.5	How to Delete Selected Dates from a Calendar	55
4.3	PJS Event Commands	56
4.3.1	PJEVLIST	57
4.3.1.1	How to List One Event	57
4.3.1.2	How to List All Events for an Owner	57
4.3.1.3	How to List All Events	58
4.3.2	PJEVPOST	59
4.3.2.1	How to Specify the event ID	59
4.3.3	PJEVRSET	60
4.3.3.1	How to Specify the event ID	60
5.	PJS/ISPF Interface	61
5.1	PJS Main Menu Panel	63
5.2	PJS Job Request Panel System	64
5.2.1	Job Request Menu Panel	66
5.2.2	List Job Requests Panel	68
5.2.3	Add Job Request Panel	71
5.2.4	Modify Job Request Panel	73
5.2.5	Specify Job Frequency Panel	76
5.2.6	Specify Job Request Events Panel	80
5.2.7	Delete Job Request Panel	83
5.2.8	Display Job Request Panel	86
5.2.9	Display Job Request Events Panel	89
5.2.10	Browse Saved JCL Panel	91
5.3	PJS Calendar Panel System	92
5.3.1	Calendar Menu Panel	94
5.3.2	List Calendars Panel	96
5.3.3	Add Calendar ID Panel	98
5.3.4	Add Calendar Panel	99
5.3.5	Modify Calendar Panel	101
5.3.6	Delete Calendar Panel	103
5.3.7	Display Calendar Panel	105
5.3.8	Job Requests for Calendar Panel	107
5.4	PJS Event Panel System	110
5.4.1	Event Menu Panel	112
5.4.2	List Events Panel	114
5.4.3	Post Event Panel	116
5.4.4	Reset Event Panel	118
5.4.5	Display Event Panel	120
5.4.6	Job Requests for Event Panel	122

6. Examples	125
6.1 How to Run a One Time Job	126
6.2 How to Run a Daily Job	127
6.3 How to Run a Job on the First Tuesday of Each Month	128
6.4 How to Run a Job That Depends on Another Job	129
6.5 How to Set Up Several Dependent Jobs	130
6.6 How to Run a Job Whenever CICS Terminates	132
6.7 How to Run a Weekly Job After a Daily Job	133
6.8 How to Run a Job After a Manual Operation	134
6.9 How to Reset an Event	135
6.10 How to Enable a Failed Job Request	136
A. Summary of Changes	137
A.1 Changes for Release 2.1.4	137
A.2 Changes for Release 2.1.3	137
A.3 Changes for Release 2.1	137
GNU General Public License	139
GNU Free Documentation License	145

Preface

Big jobs mean slow response. By day, most computer systems are clogged with requests for service; at night, many computers only run at a fraction of their capacity. The Personal Job Scheduler (PJS®) enables you to submit batch jobs on any day, at any time, as often as needed, under any circumstances.

Because PJS does not rely on scheduling specialists or system operators, the ability to schedule job submission is placed in your hands. When scheduling is decentralized, procedure and paperwork are eliminated, and scheduling errors decrease. With the ability to schedule big or small jobs whenever you need them, the Personal Job Scheduler helps you make full use of your computer at all times of the day or night.

This manual describes how to use PJS. It assumes that you need to create ad hoc job submissions and that you have some knowledge of IBM mainframes, TSO, ISPF/TSO, and job submission standards at your site.

If you need information on error messages and abend codes encountered during PJS use, please refer to the *Personal Job Scheduler (PJS®) Messages and Codes* manual. For more information on product installation and system options, please refer to the *Personal Job Scheduler (PJS®) Installation Guide*.

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 .
<i>italic lowercase</i>	is used to display information you must supply, for example, SEND <i>filename.txt</i> .
<u>Underscores</u>	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.

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, 2004. 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

1. Introduction

Most mainframe computers run 24 hours a day, but most people work from 9 to 5. Batch jobs that require lots of memory or the exclusive use of commonly shared resources can slow the system and keep people from finishing a day's work.

Some sites squeeze through this bottleneck by staggering hours or setting up shifts. Apart from the added expense of maintaining a facility 24 hours a day, shift work often separates people who need close communication.

Other sites buy expensive production schedulers and discover that they have to create Scheduling Departments and hire scheduling specialists to run the system. This complex and costly solution centralizes scheduling, and usually engenders new procedures and paperwork, which opens the door to errors.

The problems are easy to define: there are jobs that have to run when few people are using the system; there are jobs that have to run when you're not there to submit them; there are jobs that run only after other jobs are done.

The solution is the Personal Job Scheduler (PJS®). PJS decentralizes scheduling, and gives you complete control over job scheduling. It enables you to submit jobs on any day, at any time, under any circumstances.

This manual introduces PJS concepts and use in the following chapters:

- Chapter 2** **PJS System Overview** introduces PJS concepts and provides a brief description of how PJS works.
- Chapter 3** **PJS Specification Conventions** describes how to specify some PJS entities under TSO and ISPF.
- Chapter 4** **PJS/TSO Commands** describes how to approach and use the set of PJS commands under TSO. If you plan to use PJS under ISPF, you can skip this chapter.
- Chapter 5** **PJS/ISPF Interface** describes how to use PJS on the set of distributed ISPF panels.
- Chapter 6** **Examples** describes how to use PJS commands and features. Although this chapter uses PJS/TSO commands for each example, all of the actions can be replicated in PJS/ISPF.

2. PJS System Overview

The Personal Job Scheduler (PJS) is a job scheduling system that enables you to schedule your own batch jobs. This chapter introduces PJS components and concepts.

2.1 PJS Software Overview

The following subsections briefly describe the types of data PJS accepts and how PJS software components process that data.

2.1.1 The 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 three 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.

You can safely ignore the structure of these records.

2.1.2 The PJS JCL Spool

PJS can submit a job from your data set, which is called a **JCL data set**, or from the **PJS JCL Spool**. The PJS JCL Spool is an internal data set that can contain copies of the jobs in your JCL data sets.

A user who has update access to the JCL data set cannot modify the job JCL kept on the PJS JCL Spool. A job placed on the PJS JCL Spool is protected from any user who isn't authorized to update the job request.

The PJS JCL Spool is an optional feature. Your site can disable it, it can require that you use it, or it can give you a choice.

2.1.3 The PJS/TSO Interface

PJS/TSO commands can create, modify, list, and delete job requests, calendars, and events. You can issue PJS/TSO commands at the **READY** prompt, or anywhere else TSO commands are issued. These commands can be used in TSO CLISTs or REXX EXECs to further automate PJS updates.

Detailed information on PJS/TSO commands is contained in Chapter 4.

2.1.4 The PJS/ISPF Interface

PJS/ISPF panels replicate the PJS/TSO commands, but the panels are much friendlier. In addition to a simple, menu-driven panel system, the PJS/ISPF interface provides a tutorial, context-sensitive help panels, and field-level prompts.

Detailed information on PJS/TSO commands is contained in Chapter 4.

2.1.5 The PJS System Task

The PJS System Task monitors the contents of the PJS Request Queue, takes care of events, keeps track of when each job request is to be submitted, submits the job for execution, and handles processing errors. The PJS System Task should run whenever the system runs: the system operator should start the PJS System Task at IPL time and should only stop the System Task during system maintenance.

Detailed information on the PJS System Task is contained in Section 2.6.

2.2 Owners

Because PJS can keep records for many users on a PJS Request Queue, PJS must be able to identify who owns each record. PJS associates an *owner-ID* with each record on the PJS Request Queue.

In most cases, the *owner-ID* is just the TSO user-ID of the person who added the record. However, if a record is associated with more than one owner, a site can allow an *owner-ID* to be a group-ID, a system-ID, or some arbitrary value assigned by the Site Administrator.

Detailed information on how to specify an *owner-ID* is contained in Section 3.1.

2.3 PJS Job Requests

When you tell PJS you want a job to run at a later time, you have issued a job request to PJS. Among other things, you tell PJS the name of the data set that contains the job, the date and time to submit the job, how often you want the job submitted, and any events on which the job depends. PJS puts valid job requests on the PJS Request Queue.

The following subsections briefly describe the options you can specify in a job request.

2.3.1 The *request-ID*

Each job request placed on the PJS Request Queue is assigned a unique *request-ID*. The *request-ID* consists of an *owner-ID* and a *request number*, and is usually formatted as *owner-ID.req-number*. The *request number* is a number PJS assigns when the job request is added to the PJS Request Queue.

Detailed information on how to specify a *request-ID* is contained in Section 3.2.

2.3.2 Job Request Status

Each job request has an associated status:

- WAIT** means the job request is either waiting for the next date and time of job submission, or is waiting for all of its job request events to be posted.
- COMPLETE** means that all job request processing has been completed. The last date and time for job submission is past.
- ERROR** means that the PJS System Task encountered an error as it processed the job. The PJS System Task cannot process the job request until the status is changed.
- DISABLED** means that the job request has been disabled by the user. The PJS System Task cannot process the job request until the status is changed.
- SUBMIT** means that the PJS System Task is submitting the job. After job submission is complete, the job request is placed in WAIT, COMPLETE, or ERROR status.

The PJS System Task assigns job request status. You can set or change job request status with PJS/TSO command parameters, or equivalent PJS/ISPF panels.

2.3.3 The JCL Data Set

You must specify a JCL data set for each job request. This data set contains the job you want PJS to submit for execution. PJS can submit a job from the JCL data or from the PJS JCL Spool.

2.3.4 Job Submission Date and Time

Each job request maintains several dates and times to manage job submission. You will enter some of these values, and some of these values are only maintained by PJS.

2.3.4.1 Start Date and Start Time

You will enter a **Start Date** and a **Start Time**. These values define the first date and time on which you want PJS to submit the job. Every job request must have a **Start Date** and a **Start Time**.

If you specify a frequency option that conflicts with your **Start Date**, PJS first submits the job on the next available and valid date. For example, if your **Start Date** is a Monday, but you've specified that the job is only submitted on Tuesdays, PJS begins to submit the job on the first Tuesday after your **Start Date**.

2.3.4.2 End Date and End Time

You can enter an **End Date** and an **End Time** to define the last date and time on which you want PJS to submit the job. If you don't specify these values, a job you want submitted periodically, for example, every week, can be submitted forever.

If you specify a frequency option that conflicts with your **End Date**, PJS last submits the job on the nearest previous and valid date. For example, if your **End Date** is a Wednesday, but you've specified that the job is only submitted on Tuesday, PJS stops submitting the job on the Tuesday before your **End Date**.

2.3.4.3 Next Run Date and Time

The **Next Run Date and Time** is the next scheduled date and time on which PJS can submit the job. PJS calculates this value based on the **Start Date** and **Start Time**, how often you want PJS to submit the job, and the time at which PJS performs this calculation.

Unposted events can delay job submission. More information on events is contained in Section 2.5.

2.3.4.4 Last Run Date and Time

The **Last Run Date and Time** is the last date and time on which PJS submitted the job. This is the actual date and time of submission, which is not always the scheduled date and time of submission.

2.3.5 The Submit Window

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

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

The first two factors are common, and 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 causes PJS to behave as if the job was submitted. PJS resets all job request events and, if the job is to be submitted more than once, calculates the **Next Run Date and Time**.

2.3.6 Frequency

PJS provides several scheduling options, called **frequency options**, that tell PJS how often to submit the job:

- A **one-time** job is submitted only once, at the **Start Date** and **Start Time** you specify. After the job is submitted, the job request is placed in COMPLETE status.
- A **periodic frequency** specifies a regular interval between job submissions. You can choose from **MINUTES, HOURS, DAYS, WEEKS, MONTHS**, or **YEARS**. You can specify a periodic frequency as small as 1 minute or as large as 99 years.
- A **day-of-week** frequency tells PJS to submit the job on specified days of the week. You can specify any combination of days.
- The **end-of-month** frequency tells PJS to submit the job every month on the last day of the month or some number of days before the last day of the month. Because of February, you cannot use this option to submit a job more than 27 days before the end of any month.
- The **calendar** frequency tells PJS to submit the job on the dates you have selected on a PJS calendar. You must define at least one calendar before you specify a calendar as a frequency option.

More information on calendars is contained in Section 2.4.

If the system is down when the job is supposed to run, the PJS System Task will try to submit the job after the system is restarted. If the system is down for an extended period of time, a number of planned job submissions may be skipped. The number depends on the frequency option.

2.3.7 Job Request Events

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.

More information on events and job request events is contained in Section 2.5.

2.4 PJS Calendars

You can use PJS calendars to specify arbitrary sets of dates for job submission. For example, you can select the second Monday of each month, select the set of national and state holidays, or select every day except for company-defined holidays.

You must assign a unique *calendar-ID* to each calendar you place on the PJS Request Queue. The calendar-ID consists of an *owner-ID* and a *calendar name*, and is usually formatted as *owner-ID.cal-name*.

Detailed information on how to specify a *calendar-ID* is contained in Section 3.3.

2.5 PJS Events

PJS events enable you to schedule jobs based on circumstances other than the next date and time of job submission. 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.

PJS events are automatically created and deleted. An event is created when it is specified in a job request. That same event can be specified in any number of job requests. If the event is not specified in any job request, the PJS Queue Maintenance Utility, which is run periodically by the system operator, will delete the event from the PJS Request Queue.

2.5.1 The *event-ID*

You must assign a unique *event-ID* to each event you place on the PJS Request Queue. The event-ID consists of an *owner-ID* and an *event name*, and is usually formatted as *owner-ID.event-name*.

Detailed information on how to specify an *event-ID* is contained in Section 3.4.

2.5.2 Events and Job Request Events

PJS keeps event information in two different places:

- Each unique event you specify in a job request command has a record created for it in the PJS Request Queue. This record is called an **event record**. An event record refers to an event that can be specified in many job requests.
- Each job request that specifies events contains information about each event. The information is contained in the job request record and is called a **job request event**. A job request event is specific to the job request in which it is specified.

When you specify a new, unique event, PJS will create an event record and put a job request event in the job request. When you specify the same event in other job requests, PJS puts a job request event in the job request, but does not create a new event. Although events and job request events are closely related, you can treat job request events as independent entities.

The PJS System Task only uses an event record to post or reset the corresponding job request events in all of the job requests that specify the event. When an event record is posted or reset, PJS puts the event

record in POST PENDING or RESET PENDING status. The PJS System Task then posts or resets all of the corresponding job request events. After the job request events are posted or reset, the event record returns to its normal state.

When the PJS System Task checks to see if a job request is ready for submission, it checks the job request events, not the event record. If all of the job request events are posted, PJS submits the job. After submission, all of the job request's job request events 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.

2.5.3 How to Post Events or Job Request Events

When you tell PJS that the outside condition associated with the event is satisfied, you **post** the event. To post an event, you can use one of the following methods:

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

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

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.

- You can use the PJS/TSO **PJEVPOST** command or the **PJS/ISPF Post Event Panel**
- You can use a site-created MVS System Exit.

If you want to post job request events in a single job request, you can use PJS/TSO job request commands or PJS/ISPF job request panels. The corresponding event record and corresponding job request events in other job requests are not posted.

2.5.4 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.

2.5.5 How to Reset Events or Job Request Events

When you remove the posting from an event, you **reset** the event. In most cases, the only reason to reset an event is to remove the posting from events and job request events 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 a step to the job:

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

where:

stepname is any valid stepname.

owner-ID is the owner-ID of the event to be reset. You must specify this value.

event-name is the event name of the event to be reset.

- You can use the PJS/TSO **PJEVREST** command or the **PJS/ISPF Reset Event Panel**

If you want to reset job request events in a single job request, you can modify the job request.

2.6 PJS System Task Process Summary

The PJS System Task manages the PJS Request Queue and job submission.

The following diagram shows how PJS handles your job requests:

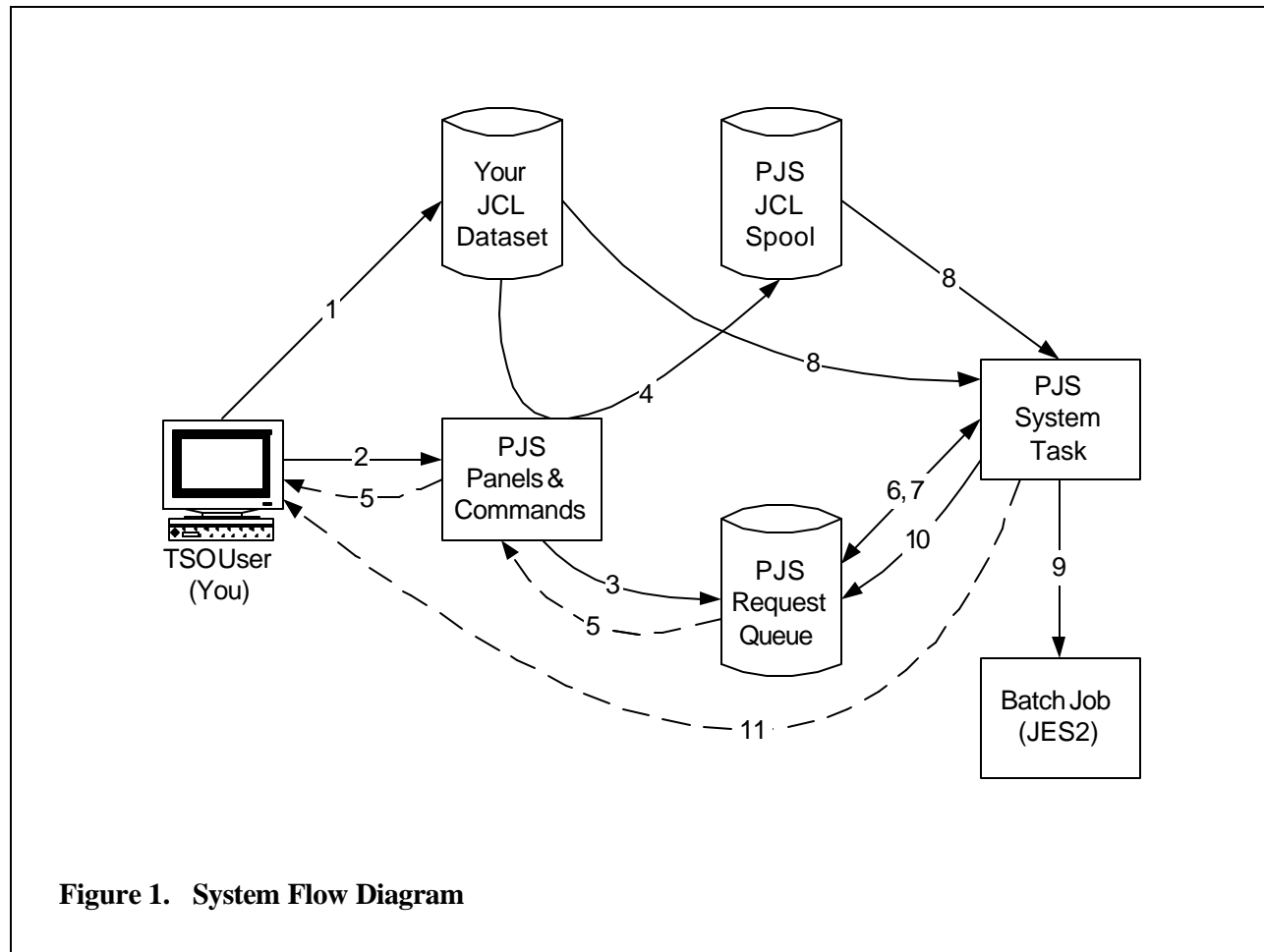


Figure 1. System Flow Diagram

To schedule a job, you should do the following:

1. Put the JCL for job you want to schedule into a data set.
2. Use the PJS ISPF panels or the PJS TSO commands to add the job to PJS.
3. The PJS ISPF panels or TSO commands will add the **Job Request** to the **PJS Request Queue**.
4. If you specified the **Save JCL** option on your Job Request, the PJS ISPF panels or TSO commands will copy your JCL from your JCL dataset to the **PJS JCL Spool**.
5. PJS ISPF panels and TSO commands are also available to query the status of your Job Requests. ISPF panels and TSO commands are also be used to modify or delete existing Job Requests, and to create, modify, and delete **PJS Calendars**, and to post or reset **PJS Events**.

About once a minute, the **PJS System Task** checks the **PJS Request Queue** for any new or modified **Job Requests**, or for any posted or reset **Events**.

6. First, if any posted or reset **Events** are found, the related **Job Request Events** may be posted or reset. For each related **Job Request Event**:
 - If the **Job Request Event** can be 'preposted', it will be posted even if posting occurs before the **Next Run Date and Time**.
 - If the **Job Request Event** cannot be 'preposted', and it will be posted only if the **Next Run Date and Time** for the Job Request has been reached. Otherwise the **Job Request Event** will not be posted.
 - If the **Event** is reset, every related **Job Request Event** is reset. The 'prepost' attribute does not affect this case.
7. Next, the **PJS System Task** checks the PJS Request Queue for Job Requests that are ready to be submitted. For each Job Request that is ready to be submitted, the System Task uses the following procedure:
 - The PJS System Task retrieves the PJS job request record from the PJS Request Queue. The status of the Job Request is changed to **SUBMIT** to lock the Job Request and prevent anyone from updating the Job Request while it is being submitted.
 - The **Next Run Date and Time** is checked against the job request's **Submit Window** and the current system date and time. If Submit Window has expired the Job Request will be either rescheduled or set to the **ERROR** status. Otherwise processing continues.
8. The JCL for the job is retrieved from the user's JCL Dataset or PJS JCL Spool.
9. The PJS System Task submits the job for execution.
10. After the job is submitted, the Job Request is updated to reflect the results of the submit:
 - Any **Job Request Events** are reset.
 - The **Last Run Date and Time** is set to the actual time of job submission.
 - A new value for the **Next Run Date and Time** is calculated.
 - If there is a new value for **Next Run Date and Time**, PJS places the job request in **WAIT** status.
 - If there is no **Next Run Date and Time**, PJS places the job request in **COMPLETE** status.
 - If an error occurred during processing, the job request is placed in **ERROR** status.

11. The PJS System Task sends a message to tell you that job submission succeeded or failed. Messages are sent by the TSO **SEND** command with the **LOGON** option. All messages are recorded in the system log.

To receive PJS messages, your TSO profile must include the **INTERCOMM** operand. If you're not logged on when a PJS message is issued, or if you've chosen not to receive messages during the course of a session, the message is saved in the TSO messages data set, and the message is sent the next time you log on.

3. PJS Specification Conventions

Some value specifications are used in many PJS commands and parameters. This manual will refer to the information in this chapter. All specifications and descriptions are valid under TSO or ISPF. Any exceptions will be discussed in Chapters 4 and 5.

The following specifications will be discussed:

- The *owner-ID* specifies who plans to work on a job request, calendar, or event. The *owner-ID* is part of the *request-ID*, the *calendar-ID*, and the *event-ID*. For more information, please refer to Section 3.1.
- The *request-ID* specifies who "owns" a job request. For more information, please refer to Section 3.2.
- The *calendar-ID* specifies who "owns" a calendar. For more information, please refer to Section 3.3.
- The *event-ID* specifies who "owns" an event. For more information, please refer to Section 3.4.
- **Time Specifications** are used to specify values such as the time of submission. For more information, please refer to Section 3.5.
- **Date Specifications** are used to specify values such as the first date of submission or the dates on a PJS calendar under TSO. For more information, please refer to Section 3.6.
- **Job Request Status Values** describe the various status values that a PJS Job Request can have. For more information, please refer to Section 3.7.

3.1 The *owner-ID*

This value specifies who "owns" a given PJS entity:

owner-ID specifies who "owns" the job request, calendar, or event on which you plan to work. Ordinarily, you will specify the *owner-ID* as part of a *request-ID*, a *calendar-ID*, or an *event-ID*.

In most cases, *owner-ID* is only a synonym for a user-ID. However, site standards can ascribe other meanings to the value of *owner-ID*; for example, your site may define an *owner-ID* as either a user-ID or a TSO group name.

Under TSO, the default is your TSO user-ID. Under ISPF, the default is the value saved in the ISPF profile. If you plan to work on a job request, a calendar, or an event under the default, you can omit this specification.

Only authorized users can work on another owner's job requests, calendars, or events. Because authorization is site-specific, your site may restrict your access to other user-IDs or *owner-IDs*.

A valid value is 1 through 8 characters long. The first character must be alphabetic. For more information on valid values you can specify for *owner-ID*, please contact your Site Administrator.

3.2 The *request-ID*

The *request-ID* identifies individual job requests. When you work with a job request, you will need to specify a *request-ID* in the *[owner-ID.]req-number* format, where:

owner-ID specifies who "owns" the job request. For more information on *owner-ID*, please refer to Section 3.1.

req-number is the number PJS assigns to the job request after it has been created. You cannot specify a *req-number* for a job request you plan to add to the PJS Request Queue.

If you plan to work on existing job requests, you must specify a *req-number*. A valid value is 1 through 3 digits long. If you have forgotten the *req-number*, you can display a list of your job requests under TSO or ISPF.

3.3 The *calendar-ID*

The *calendar-ID* identifies individual calendars. If you work with a calendar, you will need to specify a *calendar-ID* in the *[owner-ID.]cal-name* format, where:

owner-ID specifies who "owns" the job request. For more information on *owner-ID*, please refer to Section 3.1.

cal-name is the name of the calendar. A valid value is 1 through 8 characters long. The first character must be alphabetic.

If you create a calendar, you must specify a valid and distinct calendar name. If you plan to work on an existing calendar, the *cal-name* must exist.

3.4 The *event-ID*

The *event-ID* identifies individual events and job request events. If you work with an event, you will need to specify an *event-ID* in the *[owner-ID.]event-name* format, where:

owner-ID specifies who "owns" the job request. For more information on *owner-ID*, please refer to Section 3.1.

event-name is the name of the event or job request event. A valid value is 1 through 8 characters long. The first character must be alphabetic.

If you specify a new job request event, you must specify a valid and distinct event name. If you plan to work with an existing event, the *event-name* must exist.

3.5 Time Specifications

When you work with job requests, you will need to specify values such as the time you want PJS to submit a job for execution. PJS enables you to specify an **absolute time** or a **relative time**.

3.5.1 Absolute Time

An absolute time is a fully specified time, for example, 10:22 or 10:22PM.

To specify an absolute time, use the *hh:mm[AM|PM]* format:

hh is the hour. If you specify **AM** or **PM**, *hh* must be between **01** and **12**. If you do not specify **AM** or **PM**, the 24-hour clock is assumed, and *hh* must be between **00** and **23**.

mm is the minute. Valid values are **00** through **59**.

AM means that the specified time is between midnight and noon. **12:00 AM** is midnight.

PM means that the specified time is between noon and midnight. **12:00 PM** is noon.

3.5.2 Relative Time

A relative time does not require a time specification, but you do have to specify a number. That number is added to the current system time to determine the time of submission. For example, if you plan to run a job "an hour from now," you can enter a relative time value of 1.

To specify a relative time, use one of the following formats:

**+hh* to add the number of hours (*hh*) to the current system time.

**+:mm* to add the number of minutes (*mm*) to the current system time.

**+hh:mm* to add the number of hours (*hh*) plus minutes (*mm*) to the current system time.

In each of the above cases:

hh is the number of hours to be added to the current system time. Valid values are **00** through **99**.

mm is the number of minutes to be added to the current system time. Valid values are **00** through **99**.

3.6 Date Specifications

When you work with job requests, you may need to specify values such as the day on which PJS will submit a job for execution. If you use TSO calendar commands, you will need to specify date values. PJS enables you to specify an **absolute date** or a **relative date**.

3.6.1 Absolute Date

An absolute date is a fully specified date, for example, 10/22/2002. Normally PJS accepts and displays dates in U.S. format (that is *mm/dd/yyyy*). Your installation may have changed this to use the European date format (that is *dd/mm/yyyy*). If you are not sure which format your installation uses, check with the person who installed PJS.

To specify an absolute date in U.S. format, use the *mm/dd/yyyy* format. To specify an absolute date in European format, use the *dd/mm/yyyy* format. In both cases:

mm is the month. Valid values are **01** through **12**, i.e., January through December.

dd is the day of the month. Valid values are **01** through **31**.

yyyy is the year. This may be a 2-digit year, or a 4-digit year. 2-digit years are assumed to be in the 21st century, i.e. **2000** through **2099**. Valid values are **00** through **99**, or **2000** through **2099**.

The rest of this document assumes your installation uses the U.S. data format. If you are using the European data format, simply replace the U.S. format shown with the European format.

3.6.2 Relative Date

A relative date does not require a date specification, but you do have to specify a number. That number is added to the current system date to determine the day of submission. For example, if you plan to run a job "a week from today," you can enter a relative date value of 7.

To specify a relative date, use the **+ddd* format, where:

**+* indicates a relative date.

ddd is the number of days to be added to the current system date. Valid values are **001** through **999**.

3.7 Job Request Status Values

The Job Request Status indicates the current state of processing for a job request. The following are the possible values for the Job Request Status:

- WAIT** means that the job request is waiting, either for its submit time, or for its events to be posted.
- COMPLETE** means that the job request is finished. The job will not be submitted again unless the job request is changed. To change the status back to WAIT, modify the job request, entering a new **Start Date** and **Start Time** and either the **ENABLE** TSO command option, or enter **YES** in the **Enabled** input field of the **Modify Job Request** ISPF panel.
- DISABLED** means the job request is on the PJS Request Queue but is not active. This status is usually set by the user “disabling” the job request. It can also be set if the Submit Window Time for the job request was exceed, if the Window Option is also set to DISABLE. To change the status back to WAIT, modify the job request with either the **ENABLE** TSO command option, or enter **YES** in the **Enabled** input field of the **Modify Job Request** ISPF panel.
- ERROR** means that the PJS System Task encountered an error while processing this job request. To change the status back to WAIT, correct the error, then modify the job request with either the **ENABLE** TSO command option, or enter **YES** in the **Enabled** input field of the **Modify Job Request** ISPF panel.
- SUBMIT** means that the job request is being processed by the PJS System Task. The job request cannot be modified until this process completes. The user cannot change this status. If this status persists contact your PJS site administrator.

4. PJS TSO Commands

Note: If you plan to use PJS under ISPF, you can skip this chapter.

The Personal Job Scheduler (PJS) TSO command set contains three sets of commands:

- **PJS Job Request Commands**, which enable you to create, delete, list, and modify job requests. The commands are:

PJREQADD
PJREQDEL
PJREQLST
PJREQMOD

Job request commands are discussed in Section 4.1.

- **PJS Calendar Commands**, which enable you to create, delete, list, and modify calendars. The commands are:

PJCALADD
PJCALDEL
PJCALIST
PJCALMOD

Calendar commands are discussed in Section 4.2.

- **PJS Event Commands**, which enable you to list, post, and reset events. The commands are:

PJEVLST
PJEVPOST
PJEVRSET

Event commands are discussed in Section 4.3.

Commands are listed in alphabetical order. Each command description begins with a discussion of the command keyword followed by discussions of command parameters. The parameters are listed in functional order.

4.1 PJS Job Request Commands

PJS job request commands enable you to create, delete, list, and modify job requests. The TSO commands are:

- PJREQADD** which enables you to create a job request. The job request is placed on the PJS Request Queue, so PJS can submit your job for execution. To create a job request, specify the name of the data set that contains the job and the time you want the job submitted. Many other options are available. **PJREQADD** is discussed in Subsection 4.1.1.
- PJREQDEL** which enables you to delete a job request from the PJS Request Queue. **PJREQDEL** is discussed in Subsection 4.1.2.
- PJREQLIST** which displays a list of job requests. The characteristics and values of each job request are displayed. **PJREQLIST** is discussed in Subsection 4.1.3.
- PJREQMOD** which enables you to modify an existing job request. **PJREQMOD** is discussed in Subsection 4.1.4.

Commands are listed in alphabetical order. Each command description begins with a discussion of the command keyword followed by discussions of command parameters. The parameters are listed in functional order.

4.1.1 PJREQADD

Use the **PJREQADD** command to place a job request on the PJS Request Queue:

```

PJREQADD    [ OWNER(owner-ID) ]

             JCLDSN(dsname [ (member) ])

             [ NOSAVE | SAVE ]

             [ STRTDATE [ (mm/dd / [cc] yy) | (*+ddd) ] ]

             STRTTIME { (hh:mm [ AM | PM ] ) | (*+ { hh:mm | hh | :mm } ) }

             [ ONCE |
               MINUTES [ (nnn) ] |
               HOURS [ (nnn) ] |
               DAYS [ (nnn) ] |
               WEEKDAYS (day1 [ [ , day2 ] ... ] ) |
               WEEKS [ (nnn) ] |
               EOM [ (nn) ] |
               MONTHS [ (nn) ] |
               YEARS [ (nn) ] |
               CALENDARS ( [ owner-ID. ] cal-name [ , ... ] ) ]

             [ ENDDATE { (mm/dd / [cc] yy) | (*+ddd) } ]

             [ ENDTIME { (hh:mm [ AM | PM ] ) | (*+ { hh:mm | hh | :mm } ) } ]

             [ EVENTS ( [ owner-ID. ] event-name [ /NOPREPOST | /PREPOST ] [ , ... ] ) ]

             [ POSTEVENTS ( [ owner-ID. ] event-name [ , ... ] ) ]

             [ WINDOWTIME (hh:mm) ]

             [ WINDOWOPT ( [ DISABLE | ERROR | SKIP ] ) ]

             [ DISABLED | ENABLED ]

```

PJREQADD begins the **PJREQADD** command, which you can use to put a job request on the PJS Request Queue. **PJRA** is a synonym of **PJREQADD**. You can also specify **PJADD** or **PJA**: these synonyms are included for compatibility with PJS 1.0.

Use the parameters described in the remainder of this subsection to specify job request characteristics. Only two types of parameters are required:

- The **JCLDSN** parameter tells PJS the name of the data set that contains the job to be executed. For more information, please refer to **4.1.1.2 How to Specify the Data Set Name**.

- The **STRTTIME** parameter tells PJS when to submit the job for execution. For more information, please refer to **4.1.1.3 How to Specify the Time of Submission**.

Although only the **JCLDSN** and **STRTTIME** parameters are required, the optional parameters provide power and flexibility. Although some parameters have prerequisites, you can specify parameters in any order.

With one exception, any of the values specified in any of the parameters can be changed or deleted by using the **PJREQMOD** command, which is discussed in Subsection 4.1.4. The exception is the value specified in the **OWNER** parameter, which is discussed in **4.1.1.1 How to Specify an OWNER**.

4.1.1.1 How to Specify an OWNER

This parameter is optional:

OWNER specifies who will "own" the job request. The default is your current TSO user-ID. You don't have to use this parameter unless you are creating a job request for another owner.

To complete this parameter, use the (*owner-ID*) format, where:

owner-ID specifies who will "own" the job request. For more information on the *owner-ID* specification, please refer to Section 3.1.

4.1.1.2 How to Specify the Data Set Name

This parameter is required:

JCLDSN is short for **JCL Data Set Name**. This keyword tells PJS that the job request will specify a sequential data set or the member of a partitioned data set. The data set or member contains the job you want PJS to submit for execution.

To complete this parameter, specify the data set name by using the (*dsname*[(*member*)]) format, where:

dsname is either the name of a sequential data set or the prefixes for a partitioned data set. Use TSO syntax and site standards to enter this value:

If you specify '*dsname*', where *dsname* is enclosed by quotes, the *dsname* is treated as a fully qualified data set name.

If you specify *dsname*, without quotes, the TSO data set prefix is automatically prepended to *dsname* to form *prefix.dsname*. In most cases, *prefix* is your TSO user-ID.

member is the name of a member of a partitioned data set. If you typed in the name of a sequential data set, omit this specification.

DATASET and **DSNAME** are synonyms of **JCLDSN**.

4.1.1.3 How to Use the PJS JCL Spool

The optional **SAVE** or **NOSAVE** parameters determine whether you place a copy of your job JCL on the PJS JCL Spool. These keywords are mutually exclusive.

If you place job JCL on the PJS JCL Spool, PJS can provide protection from accidental modification or unauthorized access. However, if your site uses a security package such as RACF, your job JCL may already be protected from intrusive access.

The default for PJS JCL Spool use was determined when PJS was installed. Your site may require you to specify either **NOSAVE** or **SAVE**, or you may be able to specify either keyword. For more information, please contact your Site Administrator or the person who installed PJS.

SAVE tells PJS to place a copy of the job JCL on the PJS JCL Spool. When the job is submitted, the JCL on the PJS JCL Spool will be executed.

Later, if you decide to remove the job JCL from the PJS JCL spool, you can specify the **NOSAVE** keyword in the **PJREQMOD** command. If you change the job JCL and want to apply the changes to future job submissions, you can specify the **RESAVE** keyword, which replaces the old JCL with the new. For more information on the **PJREQMOD** command, please refer to Subsection 4.1.4.

NOSAVE tells PJS not to place a copy of the job JCL on the PJS JCL Spool. When the job is submitted, the JCL in a private data set or in a publicly available data set will be executed.

Later, if you decide to place a copy of the job JCL on the PJS JCL spool, you can specify the **SAVE** keyword in the **PJREQMOD** command. For more information on the **PJREQMOD** command, please refer to Subsection 4.1.4.

4.1.1.4 How to Specify the Date of Submission

This parameter is optional:

STRTDTE tells PJS which day you want to submit the job for execution. The default is the current system date, which is usually today's date.

If you decide to specify a value, you can specify an **absolute** date in the (*mm/dd/yyyy*) format or a **relative** date in the (**+ddd*) format. For more information on date specifications, please refer to Section 3.6.

DATE and **RUNDATE** are synonyms of **STRTDTE**.

4.1.1.5 How to Specify the Time of Submission

This parameter is required:

STRTTIME tells PJS what time you want to submit the job for execution. Because the time parameter is required, there is no default. You must specify a valid time.

You can specify an **absolute** time in the (*hh:mm[AM|PM]*) format or a **relative** time in the (**+{hh:mm|hh|mm}*) format. For more information on time specifications, please refer to Section 3.5.

RUNTIME and **TIME** are synonyms of **STRTTIME**.

4.1.1.6 How to Specify a Frequency Option

A frequency option specifies how often PJS is to submit the job for execution. All of the following parameters are optional. The default is **ONCE**, which means that the job is submitted only one time.

You can specify periodic submission by using one of the **MINUTES** through **YEARS** parameters, or you can specify arbitrary submissions by using the **CALENDARS** parameter. Each time the job is submitted, PJS updates the job request on the PJS Request Queue. If you select **MINUTES** or **HOURS**, PJS updates the run time and, if needed, the run date. The other parameters update the run date but not the run time.

You can only specify one of the following parameters:

ONCE means that the specified job is submitted for execution only one time. After the job is submitted, PJS places the job request in **COMPLETE** or **ERROR** status. The job request remains on the PJS Request Queue until it is deleted or modified. This is the default.

MINUTES(*nnn*) means that the job is to be submitted for execution every *nnn* minutes, where *nnn* is a value from **1** through **999**. The default is **1**.

For example, **MINUTES** means that the job is submitted every minute, and **MINUTES(30)** means that the job is submitted every half-hour.

MINS is a synonym of **MINUTES**.

HOURS(*nnn*) means that the job is to be submitted for execution every *nnn* hours, where *nnn* is a value from **1** through **999**. The default is **1**.

For example, **HOURS** means that the job is submitted every hour, and **HOURS(12)** means that the job is submitted every half-day.

DAYS(*nnn*) means that the job is to be submitted for execution every *nnn* days, where *nnn* is a value from **1** through **999**. The default is **1**.

For example, **DAYS** means that the job is submitted every day, and **DAYS(3)** means that the job is submitted every third day.

DAILY is a synonym of **DAYS**.

WEEKDAYS means that the specified job is submitted for execution every week on each specified day. To specify weekdays, use the (*day1*[[*,day2*]...]) format, where:

day is one of the following:

SUNDAY
MONDAY
TUESDAY
WEDNESDAY
THURSDAY
FRIDAY
SATURDAY

You can use any abbreviation sufficient to distinguish one day from another, for example, **TU** for Tuesday, **SA** for Saturday, or **F** for Friday.

You can specify the days in any order: **WEEKDAYS (TUESDAY,MONDAY)** is a valid statement, as is **WEEKDAYS(MONDAY,TUESDAY)**.

WKDAYS and **DOW** are synonyms for **WEEKDAYS**.

WEEKS(*nnn*) means that the job is to be submitted for execution every *nnn* weeks, where *nnn* is a value from **1** through **999**. The default is **1**.

For example, **WEEKS** means that the job is submitted every week, and **WEEKS(3)** means that the job is submitted every third week.

WEEKLY is a synonym of **WEEKS**.

EOM(*nn*) means that the job is to be submitted *nn* days before the **end of each month**, where *nn* is a value from **0** through **27**. The default is **0**.

If you specify **EOM**, the job is submitted each month on the last day of the month, for example, May 31 or June 30. If you specify **EOM(3)**, the job is submitted three days before the end of each month, for example, May 28 or June 27.

MONTHS(*nn*) means that the job is to be submitted for execution every *nn* months, where *nn* is a value from **1** through **99**. The default is **1**.

For example, if you specify **MONTHS**, the job is submitted each month. If you specify **MONTHS(3)**, the job is submitted once every three months.

Caution: If the number of days in a month is less than the value of **STRTDAT**, PJS does not submit your job during that month.

For example, if the start date for a monthly job is January 31, PJS assumes you want to submit the job on the 31st of each month and that every month has 31 days. To calculate the next run date, it adds 31 days to January 31: the next run date is February 31st. The job is actually submitted 31 days from January 31. In most years, the next actual run date is March 3, so the job is not submitted during the month of February. The job is next submitted on March 31, so the monthly job is submitted twice in March.

You can avoid this problem by specifying a start date that begins no later than the 28th day of the month. For example, if you specify **MONTHS** and a start date of January 16, PJS will submit the job on the 16th of each month. For more information on how to specify a start date with **STRTDATE**, please refer to **4.1.1.4 How to Specify the Date of Submission**.

MOS and **MONTHLY** are synonyms for **MONTHS**.

YEARS(*nn*) means that the job is to be submitted for execution every *nn* years, where *nn* is a value from **1** through **99**. The default is **1**.

For example, if you specify **YEARS**, the job is submitted each year. If you specify **YEARS(3)**, the job is submitted once every three years.

YRS is a synonym of **YEARS**.

CALENDARS means that PJS will submit the job for execution on the date or dates specified in a calendar. This enables you to specify an arbitrary set of submission dates. Before you can use this keyword, you must have created at least one calendar. For information on how to create a PJS calendar, please refer to Section 4.2.

To complete this parameter, specify at least one *calendar-ID* with the (*owner-ID*].[*cal-name*[,...]]) format, where:

owner-ID specifies who "owns" the calendar.

cal-name is the name of the calendar.

For more information on the *calendar-ID*, please refer to Section 3.3.

You can specify up to three calendars in a **CALENDARS** parameter. If you specify more than one calendar, PJS only submits the job on the dates that are selected on every calendar.

4.1.1.7 How to Specify an End Date

This parameter is optional. If you specify a frequency option, you may want to specify a date on which PJS stops submitting the job. If you specify or accept **ONCE** as a frequency option, you do not need to specify an end date.

ENDDATE tells PJS the date on which it will stop submitting the job for execution.

If you decide to specify a value, you can specify an **absolute** date in the (*mm/dd/[cc]yy*) format or a **relative** date in the (**+ddd*) format. For more information on date specifications, please refer to Section 3.6.

The value of the date must be equal to or greater than the value of the date specified or accepted for the **STRTDAT** parameter.

4.1.1.8 How to Specify an End Time

This parameter is optional:

ENDTIME tells PJS at what time it is to stop submitting the job for execution. If you plan to use the end time parameter, you must also enter the **STRTTIME** and **ENDDATE** parameters.

You can specify an **absolute** time in the (*hh:mm[AM|PM]*) format or a **relative** time in the (**+{hh:mm/hh:mm}*) format. For more information on time specifications, please refer to Section 3.5.

If the value of the **STRTDAT** and **ENDDATE** parameters are both equal to the same date, then the value specified for **ENDTIME** must be greater than the value specified for **STRTTIME**.

4.1.1.9 How to Specify Job Request Events

This parameter is optional:

EVENTS specifies at least one event. The PJS System Task will submit the job only if all events have been posted. This keyword enables you to create event dependencies that must be satisfied before a job can be submitted.

Before you can use event-related commands such as **PJEVPOST**, you must use the **PJREQADD** or **PJREQMOD** command to tell PJS that at least one job request event is specified; you do not need to define the event beforehand. Event notification can reside in a job, or you can manage event notification manually. For more information on PJS events, please refer to Section 4.3.

To complete this parameter, use the ([*owner-ID*.]*event-name*[/**PREPOST**]/**NOPREPOST**][,...]) format, where:

<i>owner-ID</i>	specifies who "owns" the event.
<i>event-name</i>	is the name of the event. The event does not have to exist. For more information, please refer to Section 3.4.
/PREPOST	means that the job request event will be posted in the job request record whenever the event is posted. PJS doesn't care when the event is posted. This is the default.
/NOPREPOST	means that the event will not be posted in the job request record if the event is posted before the job is scheduled for its next submission.

You can specify up to 50 events for a job request. If you plan to post job request events now, you can use the **POSTEVENTS** parameter, which is discussed in **4.1.1.10 How to Post Job Request Events**.

4.1.1.10 How to Post Job Request Events

This parameter is optional:

POSTEVENTS posts job request events in the current job request. You must specify a valid **EVENTS** parameter if you plan to specify **POSTEVENTS**.

Only job request events in the current job request you are creating with the **PJREQADD** command can be posted by the **POSTEVENTS** parameter; identical job request events in other job requests are not posted.

To complete this parameter, use the ([*owner-ID*.]*event-name*[,...]) format, where:

<i>owner-ID</i>	specifies who "owns" the event.
<i>event-name</i>	is the name of the event.

For more information on how to specify an *event-ID*, please refer to Section 3.4. After the job is submitted, all job request events are automatically reset. If you want PJS to resubmit the job, the job request events must be posted again.

4.1.1.11 How to Specify a Submit Window

Both **WINDOWTIME** and **WINDOWOPT** are optional parameters. However, if your system is down often, or if you plan to make use of events, you may want to use these parameters:

WINDOWTIME specifies a window of time during which PJS can submit your job. A job can be submitted late because PJS was down or because events were not posted. While the time window is open, PJS may resume operation or the events may be posted. If your job isn't submitted while the time window is open, the action specified by the **WINDOWOPT** parameter is taken. The **WINDOWOPT** parameter is discussed below.

The **WINDOWTIME** parameter requires you to specify a time relative to the value specified in the **STRTTIME** parameter to determine when the window closes. For example, if submission time is 10:00 and you want to open a time window for two hours, you can enter a relative time value of 2:00. The time window will close at 12:00. For more information on **STRTTIME**, please refer to **4.1.1.5 How to Specify the Time of Submission**.

To specify the time relative to the value entered in the **STRTTIME** parameter, use the (*hh:mm*) format, where:

hh is the number of hours. Valid values are **00** through **99**.

mm is the number of minutes. Valid values are **00** through 99.

WTIME is a synonym of **WINDOWTIME**.

WINDOWOPT tells PJS what to do if the job cannot be submitted within the time window specified by the **WINDOWTIME** parameter. You must specify a value for **WINDOWTIME** if you plan to use this parameter.

You can specify one of the following:

DISABLE places the job request in **DISABLED** status. The job request remains on the PJS Request Queue, but it is inactive. PJS cannot submit the job.

ERROR places the job request in **ERROR** status. The job request remains on the PJS Request Queue until it is modified or deleted. This is the default.

SKIP causes PJS to behave as if the job was submitted. PJS calculates a new run date and time, and resets all job request events. When the new start time value is reached, PJS will attempt to submit the job.

To reactivate a job request in **DISABLED** or **ERROR** status, you can specify the **ENABLED** keyword in the **PJREQMOD** command. If any of these actions are taken, you may want to enter a greater value for the **WINDOWTIME** parameter as well as checking to make sure that date and time specifications are still valid. For more information on the **PJREQMOD** command, please refer to Subsection 4.1.4.

WOPT is a synonym of **WINDOWOPT**.

4.1.1.12 How to Disable the Job Request

The **DISABLED** and **ENABLED** parameters are optional and mutually exclusive:

DISABLED puts the job request in **DISABLED** status. The job request is placed on the PJS Job Request Queue, but PJS cannot submit the job for execution. The job request is completely inert.

You can use this keyword to hold a place for a job request on the Job Request Queue, and to temporarily prevent job submission. To activate the job request, specify the **ENABLED** keyword in the **PJREQMOD** command. For more information on the **PJREQMOD** command, please refer to Subsection 4.1.4.

ENABLED puts the job request in **WAIT** status. The job request is placed on the PJS Job Request Queue. PJS can submit the job for execution.

Later, if you want to deactivate the job request but keep the job request on the PJS Job Request Queue for future use, you can specify the **DISABLED** keyword in the **PJREQMOD** command. For more information on the **PJREQMOD** command, please refer to Subsection 4.1.4.

ENABLED is the default.

4.1.2 PJREQDEL

Use the **PJREQDEL** command to delete a job request from the PJS Request Queue:

PJREQDEL [<i>owner-ID.</i>] <i>req-number</i>
--

PJREQDEL begins the **PJREQDEL** command, which you can use to delete a job request from the PJS Request Queue. **PJRD** is a synonym of **PJREQDEL**. You can also specify **PJDELETE**, **PJDEL**, or **PJD**: these synonyms are included for compatibility with PJS 1.0.

The only parameter is the *request-ID* which is required. The *request-ID* is a positional parameter, and must immediately follow the TSO command name.

4.1.2.1 How to Specify the *request-ID*

This parameter is required. For information on how to specify a request-ID, you can refer to Section 3.2, then use the [*owner-ID.*]*req-number* format to identify the job request you plan to modify:

owner-ID specifies who "owns" the job request on the PJS Request Queue.

req-number is the number PJS assigned to the job request you plan to modify. If you cannot remember the number, use the **PJREQDST** command, which is described in Subsection 4.1.3.

4.1.3 PJREQST

Use the **PJREQST** command to display a list of job requests on the PJS Request Queue:

```
PJREQST   ID( [owner-ID.] req-number )
           OWNER( owner-ID )
           ALL

           SUMMARY
           DETAIL
```

PJREQST begins the **PJREQST** command, which you can use to display a list of job requests and their characteristics. **PJRL** is a synonym of **PJREQST**. You can also specify **PJLIST** or **PJL**: these synonyms are included for compatibility with PJS 1.0.

If you specify **PJREQST** alone, PJS will display all of your job requests. Each listing will display the request-ID, the status of the job request, the next date and time of submission, the frequency, events information, the data set name, and the member name. To display a different set of job requests, or to specify how much information will be displayed, please refer to the following subsections:

4.1.3.1 How to List One Job Request

This parameter is optional:

ID means that you want to list just one job request. To complete this specification, use the (*owner-ID*.*req-number*) format, where:

owner-ID specifies who "owns" the job request to be listed.

req-number is the request-number that PJS assigned to the job request.

For more information on the *request-ID* specification, please refer to Section 3.2.

4.1.3.2 How to List All Job Requests For an Owner

This parameter is optional:

OWNER means that you want to list all of the job requests that belong to one owner or user. Because authorization is site-specific, your site may not allow you to list the job requests that belong to another owner-ID or user-ID. To complete this parameter specification, use the (*owner-ID*) format, where:

owner-ID specifies who "owns" the job requests you want to list. For more information on the *owner-ID* specification, please refer to Section 3.1.

The default is to list your own job requests.

4.1.3.3 How to List All Job Requests

This parameter is optional:

ALL means that you want to list all existing job requests. Because authorization is site-specific, your site may not allow you to list the job requests that belong to another owner-ID or user-ID.

4.1.3.4 How to Specify the Amount of Displayed Information

Both **DETAIL** and **SUMMARY** are optional parameters:

DETAIL displays all available information relevant to each job request. **DETAIL** displays all **SUMMARY** information, and provides much more information on submission dates and times. A listing for one job request will occupy more than one line. This is the default when the **ID** parameter is specified.

SUMMARY only displays the request-ID, the status of the job request, the next date and time of submission, the frequency, events information, the data set name, and the member name for each listed job request. On most terminals, a listing for one job request will occupy only one line. This is the default when the **ALL** or **OWNER** parameter is specified.

4.1.4 PJREQMOD

Use the **PJREQMOD** command to change an existing job request on the PJS Request Queue:

```

PJREQMOD    [owner-ID.]req-number

            [JCLDSN(dsname[(member)])]

            [NOSAVE|RESAVE|SAVE]

            [STRTDATE[(mm/dd/[cc]yy)|(*+ddd)]]

            [STRTTIME{(hh:mm[AM|PM])|(*+{hh:mm|hh|:mm})}

            [ ONCE |
              MINUTES[(nnn)] |
              HOURS[(nnn)] |
              DAYS[(nnn)] |
              WEEKDAYS(day1[,day2]...) |
              WEEKS[(nnn)] |
              EOM[(nn)] |
              MONTHS[(nn)] |
              YEARS[(nn)] |
              CALENDARS([owner-ID.]cal-name[,...]) |
              ADDCALENDARS([owner-ID.]cal-name[,...]) |
              DELCALENDARS([owner-ID.]cal-name[,...]) ]

            [ENDDATE{(mm/dd/[cc]yy)|(*+ddd)}|NOENDDATE]

            [ENDTIME{(hh:mm[AM|PM])|(*+{hh:mm|hh|:mm})}|NOENDTIME

            [ EVENTS([owner-ID.]event-name
                                [/NOPREPOST|/PREPOST][,...]) |
              ADDEVENTS([owner-ID.]event-name
                                [/NOPREPOST|/PREPOST][,...]) |
              DELEVENTS([owner-ID.]event-name[,...]) |
              NOEVENTS ]

            [POSTEVENTS([owner-ID.]event-name[,...])

            [RESETEVENTS([owner-ID.]event-name[,...])]

            [NOWINDOW|WINDOWTIME(hh:mm)]

            [WINDOWOPT([DISABLE|ERROR|SKIP])]

            [DISABLED|ENABLED]

```

PJREQMOD begins the **PJREQMOD** command, which you can use to modify an existing job request on the PJS Request Queue. **PJRM** is a synonym of **PJREQMOD**. You can also specify **PJMODIFY**, **PJMOD**, or **PJM**: these synonyms are included for compatibility with PJS 1.0.

All parameters are optional, except for the *request-ID*. However, to modify the job request, you must specify at least one of the other parameters.

The *request-ID* is a positional parameter, and must immediately follow the TSO command name. You can specify the other parameters in any order.

4.1.4.1 How to Specify the *request-ID*

This parameter is required. For information on how to specify a request-ID, you can refer to Section 3.2, then use the [*owner-ID*.]*req-number* format to identify the job request you plan to modify:

owner-ID specifies who "owns" the job request on the PJS Request Queue.

req-number is the number PJS assigned to the job request you plan to modify. If you cannot remember the number, use the **PJREQLST** command, which is described in Subsection 4.1.3.

4.1.4.2 How to Modify the Data Set Name

This parameter is optional:

JCLDSN specifies a new data set. The current data set specification is replaced.

JCLDSN is short for **JCL Data Set Name**. This keyword tells PJS that the job request will specify a sequential data set or the member of a partitioned data set. The data set or member contains the job you want PJS to submit for execution.

To complete this parameter, specify the data set name by using the (*dsname*[(*member*)]) format, where:

dsname is either the name of a sequential data set or the prefixes for a partitioned data set. Use TSO syntax and site standards to enter this value:

If you specify '*dsname*', where *dsname* is enclosed by quotes, the *dsname* is treated as a fully qualified data set name.

If you specify *dsname*, without quotes, the TSO data set prefix is automatically prepended to *dsname* to form *prefix.dsname*. In most cases, *prefix* is your TSO user-ID.

member is the name of a member of a partitioned data set. If you typed in the name of a sequential data set, omit this specification.

DATASET and **DSNAME** are synonyms of **JCLDSN**.

If you change the JCL Data Set, and previously specified the **SAVE** option to place a copy of job JCL on the PJS JCL Spool, you may need to specify **RESAVE** to replace a copy of the current job JCL with new job JCL. For more information, please refer to **4.1.4.3 How to Use the PJS JCL Spool**.

4.1.4.3 How to Use the PJS JCL Spool

SAVE, **NOSAVE**, and **RESAVE** are optional parameters that determine whether a copy of your job JCL is placed on the PJS JCL Spool.

If you place job JCL on the PJS JCL Spool, PJS can provide protection from accidental modification or unauthorized access. However, if your site uses a security package such as RACF, your job JCL may already be protected from intrusive access.

The default for PJS JCL Spool use was determined when PJS was installed. Your site may require you to specify either **NOSAVE** or **SAVE**, or you may be able to specify either keyword. If you cannot specify **SAVE**, you cannot specify **RESAVE**. For more information, please contact your Site Administrator or the person who installed PJS.

SAVE	tells PJS to place a copy of the job JCL on the PJS JCL Spool. When the job is submitted, the JCL saved in the PJS JCL Spool will be executed.
NOSAVE	tells PJS not to place a copy of the job JCL on the PJS JCL Spool. When the job is submitted, the JCL will be submitted directly from the data set specified by the JCLDSN parameter.
RESAVE	tells PJS to replace the copy of the job JCL on the PJS JCL Spool. If you specify RESAVE for a job request that does not have a copy of job JCL saved, PJS will display an error message.

4.1.4.4 How to Specify a New Date of Submission

This parameter is optional:

STRTDAT tells PJS which day you want to submit the job for execution. The current **STRTDAT** value is replaced.

If you decide to specify a value, you can specify an **absolute** date in the (*mm/dd/[cc]yy*) format or a **relative** date in the (**+ddd*) format. For more information on date specifications, please refer to Section 3.6.

DATE and **RUNDATE** are synonyms of **STRTDAT**.

4.1.4.5 How to Specify a New Time of Submission

This parameter is optional:

STRTTIME tells PJS what time you want to submit the job for execution. The current **STRTTIME** value is replaced.

You can specify an **absolute** time in the (*hh:mm[AM|PM]*) format or a **relative** time in the (**+{hh:mm|hh;mm}*) format. For more information on time specifications, please refer to Section 3.5.

RUNTIME and **TIME** are synonyms of **STRTIME**.

4.1.4.6 How to Specify a New Frequency Option

A frequency option specifies how often PJS is to submit the job for execution. All of the following parameters are optional.

You can specify periodic submission by using one of the **MINUTES** through **YEARS** parameters, or you can specify arbitrary submissions by using the **CALENDARS** parameter. Each time the job is submitted, PJS updates the job request on the PJS Request Queue. If you select **MINUTES** or **HOURS**, PJS updates the run time and, if needed, the run date. The other parameters update the run date but not the run time.

In most cases, the current frequency option is replaced. Only the **ADDCALENDARS** and **DELCALENDARS** parameters modify, but do not replace the current option.

You can only specify one of the following parameters:

ONCE means that the specified job is submitted for execution only one time. After the job is submitted, PJS places the job request in **COMPLETE** or **ERROR** status. The job request remains on the PJS Request Queue until it is deleted or modified. This is the default.

MINUTES(*nnn*) means that the job is to be submitted for execution every *nnn* minutes, where *nnn* is a value from **1** through **999**. The default is **1**.

For example, **MINUTES** means that the job is submitted every minute, and **MINUTES(30)** means that the job is submitted every half-hour.

MINS is a synonym of **MINUTES**.

HOURS(*nnn*) means that the job is to be submitted for execution every *nnn* hours, where *nnn* is a value from **1** through **999**. The default is **1**.

For example, **HOURS** means that the job is submitted every hour, and **HOURS(12)** means that the job is submitted every half-day.

DAYS(*nnn*) means that the job is to be submitted for execution every *nnn* days, where *nnn* is a value from **1** through **999**. The default is **1**.

For example, **DAYS** means that the job is submitted every day, and **DAYS(3)** means that the job is submitted every third day.

DAILY is a synonym of **DAYS**.

WEEKDAYS

means that the specified job is submitted for execution every week on each specified day. To specify weekdays, use the (*day1*[[*,day2*]...]) format, where:

day is one of the following:

SUNDAY
MONDAY
TUESDAY
WEDNESDAY
THURSDAY
FRIDAY
SATURDAY

You can use any abbreviation sufficient to distinguish one day from another, for example, **TU** for Tuesday, **SA** for Saturday, or **F** for Friday.

You can specify the days in any order: **WEEKDAYS (TUESDAY,MONDAY)** is a valid statement, as is **WEEKDAYS(MONDAY,TUESDAY)**.

WKDAYS and **DOW** are synonyms for **WEEKDAYS**.

WEEKS(*nnn*)

means that the job is to be submitted for execution every *nnn* weeks, where *nnn* is a value from **1** through **999**. The default is **1**.

For example, **WEEKS** means that the job is submitted every week, and **WEEKS(3)** means that the job is submitted every third week.

WEEKLY is a synonym of **WEEKS**.

EOM(*nn*)

means that the job is to be submitted *nn* days before the **end of each month**, where *nn* is a value from **0** through **27**. The default is **0**.

If you specify **EOM**, the job is submitted each month on the last day of the month, for example, May 31 or June 30. If you specify **EOM(3)**, the job is submitted three days before the end of each month, for example, May 28 or June 27.

MONTHS(*nn*)

means that the job is to be submitted for execution every *nn* months, where *nn* is a value from **1** through **99**. The default is **1**.

For example, if you specify **MONTHS**, the job is submitted each month. If you specify **MONTHS(3)**, the job is submitted once every three months.

Caution: If the number of days in a month is less than the value of **STRTDATE**, PJS does not submit your job during that month.

For example, if the start date for a monthly job is January 31, PJS assumes you want to submit the job on the 31st of each month and that every month has 31 days. To calculate the next run date, it adds 31 days to January 31: the next run date is February 31st. The job is actually submitted 31 days from January 31. In most years, the next actual run date is March 3, so the job is not submitted during the

month of February. The job is next submitted on March 31, so the monthly job is submitted twice in March.

You can avoid this problem by specifying a start date that begins no later than the 28th day of the month. For example, if you specify **MONTHS** and a start date of January 16, PJS will submit the job on the 16th of each month. For more information on how to specify a start date with **STRTDAT**, please refer to **4.1.4.4 How to Modify the Date of Submission**

MOS and **MONTHLY** are synonyms for **MONTHS**.

YEARS(*nn*)

means that the job is to be submitted for execution every *nn* years, where *nn* is a value from **1** through **99**. The default is **1**.

For example, if you specify **YEARS**, the job is submitted each year. If you specify **YEARS(3)**, the job is submitted once every three years.

YRS is a synonym of **YEARS**.

CALENDARS

means that PJS will submit the job for execution on the day or days specified in a calendar. This enables you to specify an arbitrary set of submission dates. Before you can use this keyword, you must have created at least one calendar. For information on how to create a PJS calendar, please refer to Section 4.2.

To complete this parameter, specify at least one *calendar-ID* with the ([*owner-ID*.]*cal-name*[,...]) format, where:

owner-ID specifies who "owns" the calendar.

cal-name is the name of the calendar.

For more information on the *calendar-ID*, please refer to Section 3.3. You can specify up to three calendars in a **CALENDARS** parameter. If you specify more than one calendar, PJS only submits the job on the dates that are selected on every calendar.

ADDCALENDARS

adds at least one more calendar to the calendars already specified for the job request. Before you can use this command, you must have at least one calendar specified for the job request. If you have not already specified a calendar, you can use the **CALENDARS** parameter to add one. This parameter assumes that at least two calendars exist and are available for your use. For information on how to create a PJS calendar, please refer to Section 4.2.

To complete this parameter, specify at least *calendar-ID* with the ([*owner-ID*.]*cal-name*[,...]) format, where:

owner-ID specifies who "owns" the calendar.

cal-name is the name of the calendar.

For more information on the *calendar-ID*, please refer to Section 3.3. You can specify up to two calendars in an **ADDCALENDARS** parameter. Each job request can have up to three calendars specified. If you specify more than one calendar, PJS only submits the job on the dates that are selected on every calendar.

DELCALENDARS deletes at least one calendar from the job request but leaves at least one calendar still specified for the job request. Before you can use this command, you must have at least two calendars specified for the job request.

To complete this parameter, specify at least *calendar-ID* with the ([*owner-ID*.]*cal-name*[,...]) format, where:

owner-ID specifies who "owns" the calendar.

cal-name is the name of the calendar.

For more information on the *calendar-ID*, please refer to Section 3.3. You can specify up to two calendars in a **DELCALENDARS** parameter. However, at least one calendar must remain specified for the job request. If you wish to delete all the calendars for a job request, you should specify a different frequency option instead.

4.1.4.7 How to Specify, Replace, or Delete an End Date

These parameters are optional:

ENDDATE tells PJS the date on which it will stop submitting the job for execution.

If you decide to specify a value, you can specify an **absolute** date in the (*mm/dd/cc*[*yy*]) format or a **relative** date in the (**+ddd*) format. For more information on date specifications, please refer to Section 3.6.

The value of the date must be equal to or greater than the value of the date specified or accepted for the **STRTDAT** parameter.

NOENDDATE removes the **ENDDATE** value previously specified in a **PJREQADD** or **PJREQMOD** command. PJS will continue to submit the job for execution until you delete the job request.

You cannot specify this parameter unless the job request has a value for **ENDDATE**. If the request has a value for **ENDTIME** and you specify **NOENDDATE**, you should also specify **NOENDTIME**. However, if you specify **NOENDTIME**, you do not also have to specify **NOENDDATE**.

ENDTIME and **NOENDTIME** are discussed in **4.1.4.8 How to Specify, Replace, or Delete an End Time**.

4.1.4.8 How to Specify, Replace, or Delete an End Time

These parameters are optional:

ENDTIME tells PJS at what time it is to stop submitting the job for execution. If you plan to use the end time parameter, you must also enter the **STRTTIME** and **ENDDATE** parameters.

You can specify an **absolute** time in the (*hh:mm*[**AM**|**PM**]) format or a **relative** time in the (*+{*hh:mm*|*hh:mm*}) format. For more information on time specifications, please refer to Section 3.5.

If the values of the **STRTDATE** and **ENDDATE** parameters are both equal to the same date, then the value specified for **ENDTIME** must be greater than the value specified for **STRTTIME**.

NOENDTIME removes the **ENDTIME** value previously specified in a **PJREQADD** or **PJREQMOD** command. There is no time at which PJS will stop submitting the job for execution.

You cannot specify this parameter unless the job request has a value for **ENDTIME**. If you specify **NOENDDATE**, you should also specify **NOENDTIME**.

4.1.4.9 How to Specify, Add, or Delete Job Request Events

These parameters are optional. To specify or replace all job request events, use **EVENTS**. To add job request events to existing job request events, use **ADDEVENTS**. To delete job request events from existing job requests, use **DELEVENTS**. To delete all job request events, use **NOEVENTS**.

EVENTS specifies the events for this job request. Any existing events are replaced with those specified. The PJS System Task will submit the job only if all events have been posted. This keyword enables you to create event dependencies that must be satisfied before a job can be submitted.

Before you can use event-related commands such as **PJEVPOST**, you must use the **PJREQADD** or **PJREQMOD** command to tell PJS that at least one job request event is specified. You do not need to define the event beforehand. Event notification can reside in a job, or you can manage event notification manually. For more information on PJS events, please refer to Section 4.3.

To complete this parameter, use the (*owner-ID*.)*event-name*[/**PREPOST**|/**NOPREPOST**][,...]) format, where:

owner-ID specifies who "owns" the event.

event-name is the name of the event. The event does not have to exist. For more information, please refer to Section 3.4.

/PREPOST means that the job request event will be posted in the job request record whenever the event is posted. PJS doesn't care when the event is posted. This is the default.

/NOPREPOST means that the event will not be posted in the job request record if the event is posted before the job is scheduled for its next submission.

You can specify up to 50 events for a job request. If you plan to post job request events now, you can use the **POSTEVENTS** parameter, which is discussed in **4.1.4.10 How to Post Job Request Events**.

ADDEVENTS adds additional events to the events already specified for the job request. For more information on PJS events, please refer to Section 4.3.

To complete this parameter, use the ([*owner-ID*].*event-name*[/**PREPOST**]/**NOPREPOST**][,...]) format, where:

owner-ID specifies who "owns" the event.

event-name is the name of the event. The event does not have to exist. For more information, please refer to Section 3.4.

/PREPOST means that the job request event will be posted in the job request record whenever the event is posted. PJS doesn't care when the event is posted. This is the default.

/NOPREPOST means that the event will not be posted in the job request record if the event is posted before the job is scheduled for its next submission.

You can specify up to 50 events for a job request. If you plan to post job request events now, you can use the **POSTEVENTS** parameter, which is discussed in **4.1.4.10 How to Post Job Request Events**.

DELEVENTS removes individual events from the job request. For more information on PJS events, please refer to Section 4.3.

To complete this parameter, specify at least *event-ID* with the ([*owner-ID*].*event-name*[,...]) format, where:

owner-ID specifies who "owns" the event.

event-name is the name of the event. The event must be specified on the job request.

For information on how to specify an *event-ID*, please refer to Section 3.4.

NOEVENTS removes all the events from the job request. The job request will not be event-dependent, only time-dependent: PJS will submit the job for execution whenever the system reaches the calculated value for the date and time of submission.

4.1.4.10 How to Post Job Request Events

This parameter is optional:

POSTEVENTS posts job request events in the current job request. Only job request events in the current job request can be posted by the **POSTEVENTS** parameter. Identical job request events in other job requests are not posted.

To complete this parameter, specify at least *event-ID* with the ([*owner-ID*.]*event-name*[,...]) format, where:

owner-ID specifies who "owns" the event.

event-name is the name of the event. The event must be specified on the job request.

For information on how to specify an *event-ID*, please refer to Section 3.4.

Posting an already posted job request event has no effect, and no error is generated.

After the job is submitted, all job request events are automatically reset. If you want PJS to resubmit the job, the job request events must be posted again.

4.1.4.11 How to Reset Job Request Events

This parameter is optional:

RESETEVENTS resets job request events in the current job request. Only job request events in the current job request can be reset by the **RESETEVENTS** parameter. Identical job request events in other job requests are not reset.

To complete this parameter, specify at least *event-ID* with the ([*owner-ID*.]*event-name*[,...]) format, where:

owner-ID specifies who "owns" the event.

event-name is the name of the event. The event must be specified on the job request.

For information on how to specify an *event-ID*, please refer to Section 3.4.

Resetting a job request event that has not been posted has no effect, and no error is generated.

4.1.4.12 How to Specify, Replace, or Delete a Submit Window

These parameters are optional:

WINDOWTIME specifies a window of time during which PJS can submit your job. A job can be submitted late because PJS was down or because events were not posted. While the time window is open, PJS may resume operation or the events may be posted. If your job isn't submitted while the time window is open, the action specified by the **WINDOWOPT** parameter is activated. The **WINDOWOPT** parameter is discussed below.

The **WINDOWTIME** parameter requires you to specify a time relative to the value specified in the **STRTIME** parameter to determine when the window closes. For example, if submission time is 10:00 and you want to open a time window for two hours, you can enter a relative time value of 2:00. The time window will close at 12:00. For more information on **STRTIME**, please refer to **4.1.4.5 How to Specify the Time of Submission**.

To specify the time relative to the value entered in the **STRTIME** parameter, use the (*hh:mm*) format, where:

hh is the number of hours. Valid values are **00** through 99.

mm is the number of minutes. Valid values are **00** through 99.

WTIME is a synonym of **WINDOWTIME**.

WINDOWOPT tells PJS what to do if the job cannot be submitted within the time window specified by the **WINDOWTIME** parameter. You must specify a value for **WINDOWTIME** if you plan to use this parameter.

You can specify one of the following:

DISABLE places the job request in DISABLED status. The job request remains on the PJS Request Queue, but it is inactive. PJS cannot submit the job.

ERROR places the job request in ERROR status. The job request remains on the PJS Request Queue until it is modified or deleted. This is the default.

SKIP causes PJS to behave as if the job was submitted. PJS calculates a new run date and time, and resets all job request events. When the new start time value is reached, PJS will attempt to submit the job.

To reactivate a job request in DISABLED or ERROR status, you can specify the **ENABLED** keyword in the **PJREQMOD** command. If any of these actions are taken, you may want to enter a greater value for the **WINDOWTIME** parameter as well as checking to make sure that date and time specifications are still valid. For more information on the **PJREQMOD** command, please refer to Subsection 4.1.4.

WOPT is a synonym of **WINDOWOPT**.

NOWINDOW removes the time window from the job request.

ENABLED puts the job request in WAIT status. The job request is placed on the PJS Job Request Queue. PJS can submit the job for execution.

4.1.4.13 How to Enable or Disable the Job Request

The **DISABLED** and **ENABLED** parameters are optional and mutually exclusive:

DISABLED puts the job request in DISABLED status. The job request is placed on the PJS Job Request Queue, but PJS cannot submit the job for execution. The job request is completely inert.

You can use this keyword to hold a place for a job request on the Job Request Queue, and to temporarily prevent job submission. To activate the job request, specify the **ENABLED** keyword.

ENABLED puts the job request in WAIT status. The job request is placed on the PJS Job Request Queue. PJS can submit the job for execution.

You can use this keyword to reactivate a job request that has been placed in DISABLED or ERROR status.

4.2 PJS Calendar Commands

PJS calendar commands enable you to create, delete, list, and modify calendars. The TSO commands are:

- PJCALADD** which enables you to create a calendar. To create a calendar, select the dates on which you plan to have PJS submit jobs for execution. You must use this command before you specify a calendar in a job request. **PJCALADD** is discussed in Subsection 4.2.1.
- PJCALDEL** which enables you to delete a calendar. **PJCALDEL** is discussed in Subsection 4.2.2.
- PJCALIST** which displays a list of calendars. The characteristics and values of each calendar are displayed. **PJCALIST** is discussed in Subsection 4.2.3.
- PJCALMOD** which enables you to modify an existing calendar. **PJCALMOD** is discussed in Subsection 4.2.4.

Commands are listed in alphabetical order. Each command description begins with a discussion of the command keyword followed by discussions of command parameters. The parameters are listed in functional order.

4.2.1 PJCALADD

Use the **PJCALADD** command to create a PJS calendar:

```
PJCALADD    [owner-ID.]cal-name

            { DATES(date|date1:date2[,...]) |
              EXCLUDE(date|date1:date2[,...]) }
```

PJCALADD begins the **PJCALADD** command, which you can use to create a PJS calendar. **PJCA** is a synonym of **PJCALADD**.

You must use this command to create a calendar before you can specify that a job request will use a calendar. After you create a calendar, you can use the **CALENDARS** parameter of the **PJREQADD** or **PJREQMOD** command. PJS will submit the job for execution on each of the dates specified in the calendar. For more information on these commands, please refer to Section 4.1.

The *calendar-ID* is a positional parameter, and must immediately follow the TSO command name. In addition, you must also specify either the **DATES** or the **EXCLUDE** parameter.

4.2.1.1 How to Specify the *calendar-ID*

This parameter is required. For more information on how to specify a *calendar-ID*, please refer to Section 3.3, then use the [owner-ID.]cal-name format, where:

owner-ID specifies who "owns" the calendar.

cal-name is the name of the calendar.

4.2.1.2 How to Select Dates by Inclusion

This parameter is not required, but you must specify either the **DATES** or the **EXCLUDE** parameter.

DATES specifies the dates to be included in the calendar. This parameter assumes that each date from the current system date on is not selected unless it is included.

To specify the dates you wish to select, use the (*date|date1:date2[,...]*) format, which enables you to specify single dates or ranges of dates. Each date specified can be an **absolute** date in the (*mm/dd/[cc]yy*) format or a **relative** date in the (**+ddd*) format. For more information on date specifications, please refer to Section 3.6.

INCLUDE is a synonym for **DATES**.

4.2.1.3 How to Select Dates by Exclusion

This parameter is not required, but you must specify either the **DATES** or the **EXCLUDE** parameter.

EXCLUDE specifies the dates to be excluded from the calendar. This parameter assumes that each date from the current system date on is selected unless it is excluded.

To specify the dates you wish to exclude, use the *(date|date1:date2[,...])* format, which enables you to specify single dates or ranges of dates. Each date specified can be an **absolute** date in the *(mm/dd/[cc]yy)* format or a **relative** date in the *(*+ddd)* format. For more information on date specifications, please refer to Section 3.6.

4.2.2 PJCALDEL

Use the **PJCALDEL** command to delete a PJS calendar:

PJCALDEL [<i>owner-ID.</i>] <i>cal-name</i>
--

PJCALDEL begins the **PJCALDEL** command, which you can use to delete a PJS calendar. **PJCD** is a synonym of **PJCALDEL**.

Deleting the calendar can affect every job request in which that calendar is specified. If you delete the calendar, check the job requests in which the calendar was specified. You may need to use the **PJREQMOD** command to ensure that PJS will continue to submit the job.

If you want to delete a calendar specification from a job request and leave the calendar intact, use the **PJREQMOD** command with either the **DELCALENDARS** parameter, or a different frequency option, such as **DAYS(*nm*)** or **EOM(*m*)**. A new frequency specification automatically deletes the current calendar specification. For more information on **PJREQMOD**, please refer to Subsection 4.1.4.

The only parameter is the *calendar-ID*, which is required. The *calendar-ID* is a positional parameter, and must immediately follow the TSO command name.

4.2.2.1 How to Specify the *calendar-ID*

This parameter is required. For more information on how to specify a *calendar-ID*, please refer to Section 3.3, then use the [*owner-ID.*]*cal-name* format, where:

owner-ID specifies who "owns" the calendar.

cal-name is the name of the calendar.

4.2.3 PJCALIST

Use the **PJCALIST** command to display a list of PJS Calendars:

```
PJCALIST    ID( [owner-ID.] cal-name )
              OWNER( owner-ID )
              ALL

              SUMMARY
              DETAIL
```

PJCALIST begins the **PJCALIST** command, which you can use to display a list of calendars and their characteristics. **PJCL** is a synonym of **PJCALIST**.

If you specify **PJCALIST** alone, PJS will display all of your calendars. Of the dates selected on each calendar, only the next date selected will be displayed. To display a different set of calendars, or to specify how much date information will be displayed, please refer to the following subsections:

4.2.3.1 How to List One Calendar

This parameter is optional:

ID means that you want to list just one calendar. To complete this specification, specify a *calendar-ID* in the (*owner-ID*.*cal-name*) format, where:

owner-ID specifies who "owns" the calendar to be listed.

cal-name is the name of the calendar.

For more information on the *calendar-ID* specification, please refer to Section 3.3.

4.2.3.2 How to List All Calendars for an Owner

This parameter is optional:

OWNER means that you want to list all of the calendars that belong to one owner or user. Because authorization is site-specific, your site may not allow you to list the calendars that belong to another owner-ID or user-ID. To complete this parameter specification, use the (*owner-ID*) format, where:

owner-ID specifies who "owns" the calendars you want to list. For more information on the *owner-ID* specification, please refer to Section 3.1.

The default is to list your own calendars.

4.2.3.3 How to List All Calendars

This parameter is optional:

ALL means that you want to list all existing calendars. Because authorization is site-specific, your site may not allow you to list the calendars that belong to another owner-ID or user-ID.

4.2.3.4 How to Specify the Amount of Displayed Information

Both **DETAIL** and **SUMMARY** are optional parameters:

DETAIL displays every date selected on each calendar. A listing for one calendar can occupy more than one line. This is the default when the **ID** parameter is specified.

SUMMARY only displays the next date selected on each calendar. On most terminals, a listing for one job request will occupy only one line. This is the default when the **ALL** or **OWNER** parameter is specified.

4.2.4 PJCALMOD

Use the **PJCALMOD** command to modify a PJS calendar:

```
PJCALMOD    [owner-ID.]cal-name

            { DATES(date|date1:date2[,...]) |
              EXCLUDE(date|date1:date2[,...]) |
              ADDDATE(date|date1:date2[,...]) |
              DELDATE(date|date1:date2[,...]) }
```

PJCALMOD begins the **PJCALMOD** command, which you can use to modify a PJS calendar by adding, deleting, or replacing date specifications. **PJCA** is a synonym of **PJCALADD**.

You must specify an existing *calendar-ID* and one of the date selection options: The *calendar-ID* is a positional parameter, and must immediately follow the TSO command name. You can specify the other parameters in any order.

4.2.4.1 How to Specify the *calendar-ID*

This parameter is required. For more information on how to specify a *calendar-ID*, please refer to Section 3.3, then use the *[owner-ID.]cal-name* format, where:

owner-ID specifies who "owns" the calendar.

cal-name is the name of the calendar.

4.2.4.2 How to Replace All Selected Dates by Inclusion

This parameter is not required, but you must specify one the **DATES**, **EXCLUDE**, **ADDDATES**, or **DELDATES** parameters.

DATES specifies the dates to be included in the calendar. This parameter assumes that each date from the current system date on is not selected unless it is included. Any previously selected dates are replaced.

To specify the dates you wish to select, use the *(date|date1:date2[,...])* format, which enables you to specify single dates or ranges of dates. Each date specified can be an **absolute** date in the *(mm/dd/[cc]yy)* format or a **relative** date in the *(*+ddd)* format. For more information on date specifications, please refer to Section 3.6.

INCLUDE is a synonym for **DATES**.

4.2.4.3 How to Replace All Selected Dates by Exclusion

This parameter is not required, but you must specify one the **DATES**, **EXCLUDE**, **ADDDATES**, or **DELDATES** parameters.

EXCLUDE specifies the dates to be excluded from the calendar. This parameter assumes that each date from the current system date on is selected unless it is excluded. Any previously selected dates are replaced.

To specify the dates you wish to exclude, use the *(date|date1:date2[,...])* format, which enables you to specify single dates or ranges of dates. Each date specified can be an **absolute** date in the *(mm/dd/[cc]yy)* format or a **relative** date in the *(*+ddd)* format. For more information on date specifications, please refer to Section 3.6.

4.2.4.4 How to Add Selected Dates to a Calendar

This parameter is not required, but you must specify one the **DATES**, **EXCLUDE**, **ADDDATES**, or **DELDATES** parameters.

ADDDATES adds additional dates to the dates already specified for the calendar. Previously selected dates are not affected.

To specify the dates you wish to select, use the *(date|date1:date2[,...])* format, which enables you to specify single dates or ranges of dates. Each date specified can be an **absolute** date in the *(mm/dd/[cc]yy)* format or a **relative** date in the *(*+ddd)* format. For more information on date specifications, please refer to Section 3.6.

4.2.4.5 How to Remove Selected Dates from a Calendar

This parameter is not required, but you must specify one the **DATES**, **EXCLUDE**, **ADDDATES**, or **DELDATES** parameters.

DELDATES removes the specified dates from the calendar.

To specify the dates you wish to remove, use the *(date|date1:date2[,...])* format, which enables you to specify single dates or ranges of dates. Each date specified can be an **absolute** date in the *(mm/dd/[cc]yy)* format or a **relative** date in the *(*+ddd)* format. For more information on date specifications, please refer to Section 3.6.

4.3 PJS Event Commands

PJS event commands enable you to list, post, and reset events. The TSO commands are:

- PJEVLIST** which displays a list of events. The characteristics and values of each event are displayed. **PJEVLIST** is discussed in Subsection 4.3.1.
- PJEVPOST** which enables you to post an event. The event must have been specified in at least one job request by using the **PJREQADD** or **PJREQMOD** command. If the event is not posted at time of submission, PJS cannot submit the job for execution. Posting an event satisfies an event dependency in at least one job request. **PJEVPOST** is discussed in Subsection 4.3.2.
- PJEVRSET** which enables you to reset an event. This command removes the event posting in at least one job request. If the event is not posted at time of submission, PJS cannot submit the job for execution. Resetting an event can create an event dependency that must be satisfied in at least one job request. **PJEVRSET** is discussed in Subsection 4.3.3.

Commands are listed in alphabetical order. Each command description begins with a discussion of the command keyword followed by discussions of command parameters. The parameters are listed in functional order.

4.3.1 PJEVLIST

Use the **PJEVLIST** command to display a list of PJS events:

```
PJEVLIST    ID( [owner-ID.] event-name )
              OWNER( owner-ID )
              ALL
```

PJEVLIST begins the **PJEVLIST** command, which you can use to display a list of events and their characteristics. **PJEL** is a synonym of **PJEVLIST**.

If you specify **PJEVLIST** alone, PJS will display all of your events. To display a different set of events please refer to the following subsections:

4.3.1.1 How to List One Event

This parameter is optional:

ID means that you want to list just one event. To complete this specification, specify an *event-ID* in the ([*owner-ID*.]*event-name*) format, where:

owner-ID specifies who "owns" the event to be listed.

event-name is the name of the event.

For more information on the *event-ID* specification, please refer to Section 3.4.

4.3.1.2 How to List All Events for an Owner

This parameter is optional:

OWNER means that you want to list all of the events that belong to one owner or user. Because authorization is site-specific, your site may not allow you to list the events that belong to another owner-ID or user-ID. To complete this parameter specification, use the (*owner-ID*) format, where:

owner-ID specifies who "owns" the events you want to list. For more information on the *owner-ID* specification, please refer to Section 3.1.

The default is to list your own events.

4.3.1.3 How to List All Events

This parameter is optional:

ALL means that you want to list all existing events. Because authorization is site-specific, your site may not allow you to list the events that belong to another owner-ID or user-ID.

4.3.2 PJEVPOST

Use the **PJEVPOST** command to post an event:

PJEVPOST [<i>owner-ID.</i>] <i>event-name</i>
--

PJEVPOST begins the **PJEVPOST** command, which you can use to post an event. **PJEP** is a synonym of **PJEVPOST**.

You must specify an existing *event-ID*: The *event-ID* is a positional parameter, and must immediately follow the TSO command name.

This command puts the event record into POST PENDING status. After the next scan interval elapses, the PJS System Task posts the job request event in job requests that have reached their date and time of submission, and in job requests that allow preposting. If the date and time of submission was not reached, and the job request event does not allow preposting, the job request event is not posted. After the PJS System Task finishes processing, POST PENDING status is removed from the event record.

If you want to post an event for only specific job requests, you can use the **POSTEVENTS** parameter of the **PJREQMOD** command to post the job request event for each job request. For more information on this command, please refer to Subsection 4.1.4.

The only parameter is the *event-ID*, which is required. The *event-ID* is a positional parameter, and must immediately follow the TSO command name.

4.3.2.1 How to Specify the *event-ID*

This parameter is required. For more information on how to specify an *event-ID*, please refer to Section 3.4, then use the [*owner-ID.*]*event-name* format, where:

owner-ID specifies who "owns" the event.

event-name is the name of the event to be posted.

4.3.3 PJEVRESET

Use the **PJEVRSET** command to reset an event:

PJEVRSET [<i>owner-ID.</i>] <i>event-name</i>
--

PJEVRSET begins the **PJEVRSET** command, which you can use to reset an event, i.e., remove the posting from an event. **PJER** is a synonym of **PJEVRSET**.

You must specify an existing *event-ID*: The *event-ID* is a positional parameter, and must immediately follow the TSO command name.

This command puts the event record into RESET PENDING status. After the next scan interval elapses, the PJS System Task resets the job request event in every relevant job request. After the PJS System Task finishes processing, RESET PENDING status is removed from the event record.

If you want to reset an event for only specific job requests, you can use the **RESETEVENTS** parameter of the **PJREQMOD** command to reset the job request event for each job request. For more information on this command, please refer to Subsection 4.1.4.

The only parameter is the *event-ID*, which is required. The *event-ID* is a positional parameter, and must immediately follow the TSO command name.

4.3.3.1 How to Specify the *event-ID*

This parameter is required. For more information on how to specify an *event-ID*, please refer to Section 3.4, then use the [*owner-ID.*]*event-name* format, where:

owner-ID specifies who "owns" the event.

event-name is the name of the event to be reset.

5. PJS ISPF Interface

An ISPF interface is distributed with PJS. The interface consists of panels that replicate the PJS TSO commands, described in Chapter 4, and a tutorial on PJS/ISPF. Only the panels that replicate the TSO commands are illustrated.

The following diagram provides a high-level overview of PJS/ISPF:

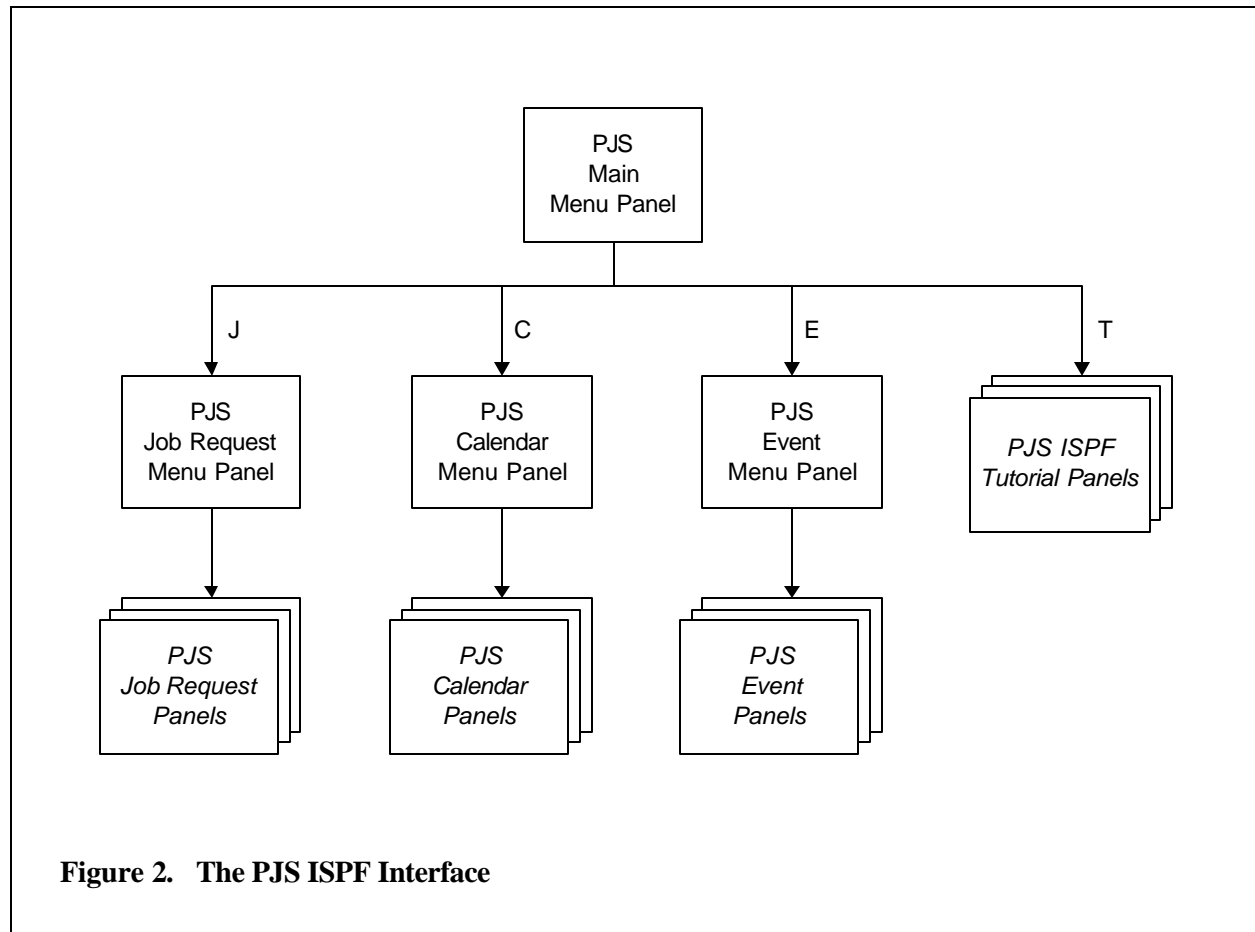


Figure 2. The PJS ISPF Interface

The **PJS Main Menu Panel** is a gateway to four panel systems. This panel is discussed in Section 5.1.

The **PJS Job Request Panel System** is entered through the **Job Request Menu Panel**. The system is described throughout Section 5.2. The menu panel is discussed in Subsection 5.2.1.

The **PJS Calendar Panel System** is entered through the **Calendar Menu Panel**. The system is described throughout Section 5.3. The menu panel is discussed in Subsection 5.3.1.

The **PJS Event Panel System** is entered through the **Event Menu Panel**. The system is described throughout Section 5.4. The menu panel is discussed in Subsection 5.4.1.

The **PJS/ISPF Tutorial Panels** are entered through the PJS Main Menu Panel or from any other panel by entering the **HELP** command. They can function as a full PJS/ISPF tutorial or as a context-sensitive help facility. If you prefer to learn PJS from online documentation, you can use the tutorial panels and skip this chapter.

Individual panels are illustrated and discussed in subsections. Each section orders its subsections functionally. In the panel illustrations, underlined fields denote input fields, that is, fields into which you can enter values. Fields without underscores are information fields.

5.1 PJS Main Menu Panel

Use the PJS Main Menu panel as a gateway to PJS/ISPF:

```
yy/mm/dd hh:mm ----- PJS(tm) - PERSONAL JOB SCHEDULER ----- v.r.m
OPTION ==> _____
```

Select one of the following functions:

```

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

T  TUTORIAL - Enter PJS/ISPF Tutorial
```

Select an option, then press ENTER.

```

(c) Copyright, Northrop Grumman, 1990, 2004.
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.
```

At most sites, you will enter PJS/ISPF from an ISPF menu panel. If you have trouble getting into PJS check with your site administrator.

The following menu options can be selected by typing the appropriate letter on the **OPTIONS** command line:

- J** is short for **Job Request**. This option will display the **PJS Job Request Menu** panel. For more information on the PJS job request system, please refer to Section 5.2.
- C** is short for **Calendar**. This option will display the **PJS Calendar Menu** panel. For more information on the PJS calendar system, please refer to Section 5.3.
- E** is short for **Event**. This option will display the **PJS Event Menu** panel. For more information on the PJS event system, please refer to Section 5.4.
- T** is short for **Tutorial**. This option will display an online tutorial on how to use PJS.

In addition, the following commands can be entered in the **OPTIONS** or **COMMAND** line of any PJS/ISPF panel:

- ABOUT** will display a panel with the PJS copyright information.
- LICENSE** will display a panel with the PJS software license.

You can exit PJS from this panel by pressing **PF3**.

5.2 PJS Job Request Panel System

The following figure shows how you can move from one panel to another in the PJS Job Request Panel System.

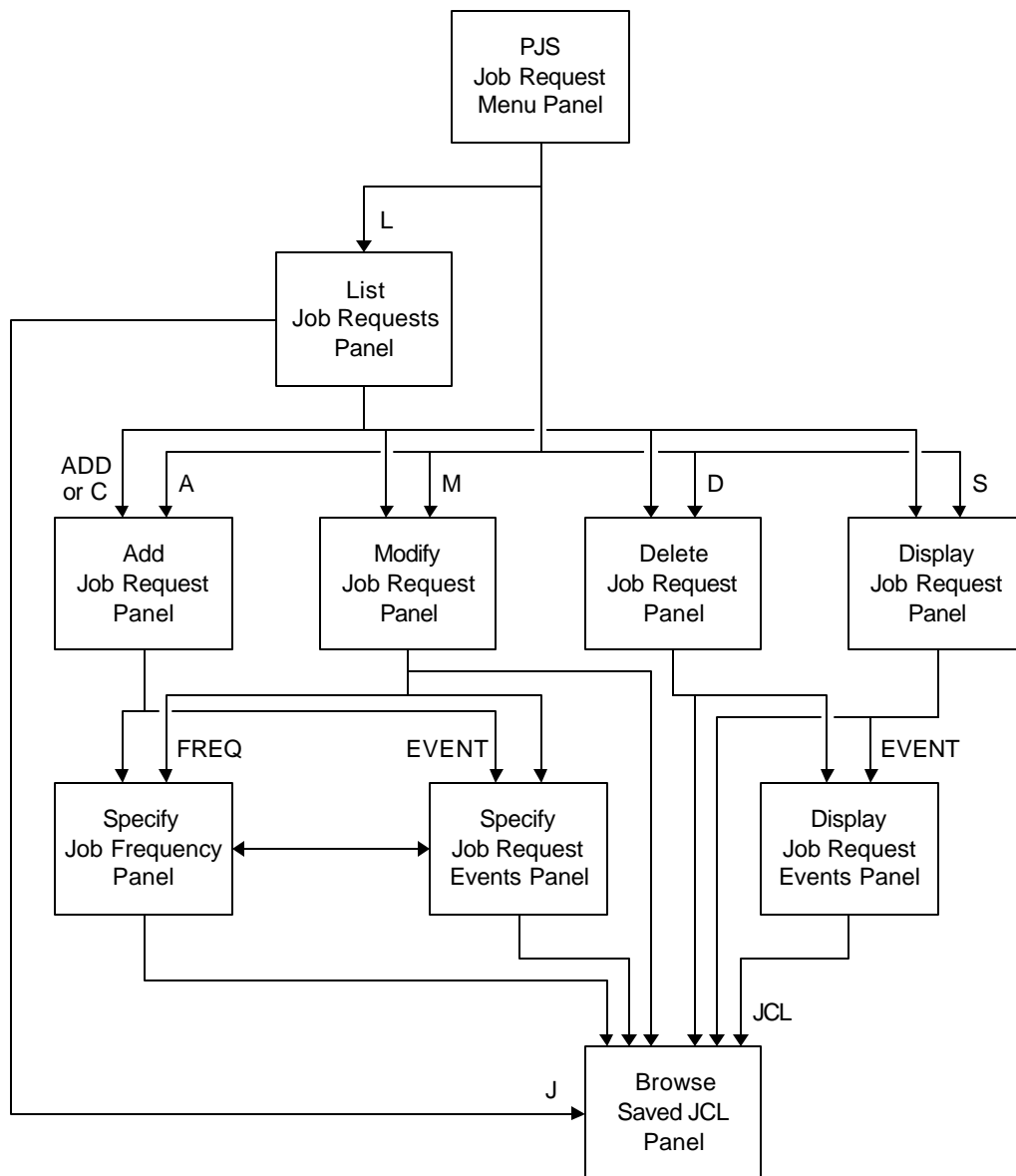


Figure 3. The PJS Job Request ISPF Panels

The **Job Request Menu Panel** is the gateway to the PJS job request panel system. From here, you can list, add, modify, delete, or display calendars. This panel is discussed in Subsection 5.2.1.

The **List Job Requests Panel** lists job requests and their characteristics. If you plan to work with more than one job request during a session, or if you have forgotten the request-ID for a job request you plan to alter, it's a good idea to use this panel. You can use line commands to select an individual job request to copy, modify, delete, or display. You can also use the **ADD** primary command to add a new job request from this panel. This panel is discussed in Subsection 5.2.2.

The **Add Job Request Panel** enables you to create a job request and place it on the PJS Request Queue. This panel is discussed in Subsection 5.2.3.

The **Modify Job Request Panel** enables you to change the characteristics of an existing job request. This panel is discussed in Subsection 5.2.4.

The **Specify Job Frequency Panel** is entered from either the **Add Job Request Panel** or the **Modify Job Request Panel**, and enables you to specify how often you want PJS to submit the job for execution. This panel is discussed in Subsection 5.2.5.

The **Specify Job Request Events Panel** is entered from either the **Add Job Request Panel** or the **Modify Job Request Panel**, and enables you to specify events, post events, and reset events for a single job request. This panel is discussed in Subsection 5.2.6.

The **Delete Job Request Panel** enables you to delete a job request from the PJS Request Queue. This panel is discussed in Subsection 5.2.7.

The **Display Job Request Panel** displays the characteristics of one PJS job request. This panel is discussed in Subsection 5.2.8.

The **Display Job Request Events Panel** is entered from either the **Delete Job Request Panel** or the **Display Job Request Panel**, and displays the name and status of all events specified for one PJS job request. This panel is discussed in Subsection 5.2.9.

The **Browse Saved JCL Panel** is entered from most of the other PJS job request panels, and displays the job JCL that is saved in the PJS Spool. This panel is discussed in Subsection 5.2.10.

5.2.1 Job Request Menu Panel

Use the Job Request Menu Panel as a gateway to the PJS job request panel system.

```
yy/mm/dd hh:mm ----- PJS(tm) - 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    ==> xxxxxxxx
Request Num ==> nnn          (Required for options D, M, and S)

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

You can enter this panel from the **PJS Main Menu Panel**

The following menu options can be selected by typing the appropriate letter on the **OPTIONS** command line:

- L** is short for **List**. This option will display the **List Job Requests** panel. This panel is described in Subsection 5.2.2. When using this option the PJS owner-ID to be listed should be entered in the **Owner-ID** input field.
- A** is short for **Add**. This option will display the **Add Job Request** panel. This panel is described in Subsection 5.2.3. When using this option the PJS owner-ID for the job request to be added should be entered in the **Owner-ID** input field.
- D** is short for **Delete**. This option will display the **Delete Job Request** panel. This panel is described in Subsection 5.2.7. When using this option the PJS job request-ID for the job request to be deleted should be entered in the **Owner-ID** and **Request Num** input fields.
- M** is short for **Modify**. This option will display the **Modify Job Request** panel. This panel is described in Subsection 5.2.4. When using this option the PJS job request-ID for the job request to be modified should be entered in the **Owner-ID** and **Request Num** input fields.
- S** is short for **Select or diSplay**. This option will display the **Display Job Request** panel. This panel is described in Subsection 5.2.8. When using this option the PJS job request-ID for the job request to be displayed should be entered in the **Owner-ID** and **Request Num** input fields.

You can exit this panel and return to the **PJS Main Menu** panel by pressing **PF3**.

The following input fields are available:

Owner-ID specifies who "owns" the job request you are adding, deleting, modifying, listing, or displaying. You must be authorized to access job requests for an owner-ID other than your own. For more information on the *owner-ID* specification, please refer to Section 3.1.

Request Num is the job request number PJS issued to a job request on the PJS Request Queue. If you plan to delete, modify, or display a job request, you must enter a value in this input field. For more information on this specification, please refer to Section 3.2.

5.2.2 List Job Requests Panel

Use the List Job Requests Panel to display a set of existing job requests.

```
yy/mm/dd hh:mm ----- PJS(tm) - LIST JOB REQUESTS ----- LINE nnnn OF nnnn
COMMAND ==> _____ SCROLL ==> xxxx

S Request-ID      Status      Next Date/Time      Frequency      Events      Member
= xxxxxxxx.nnn xxxxxxxx mm/dd/yyyy hh:mm xxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx xxxxxxxx
= xxxxxxxx.nnn xxxxxxxx mm/dd/yyyy hh:mm nnn YR NONE xxxxxxxx
= xxxxxxxx.nnn xxxxxxxx mm/dd/yyyy hh:mm SU,MO,TU,WE,TH,FR,SA nn OF nn xxxxxxxx
= xxxxxxxx.nnn xxxxxxxx mm/dd/yyyy hh:mm EOM-nn POSTED xxxxxxxx
= xxxxxxxx.nnn xxxxxxxx mm/dd/yyyy hh:mm CAL-xxxxxxxx.xxxxxxxxx xxxxxxxx xxxxxxxx
= xxxxxxxx.nnn xxxxxxxx mm/dd/yyyy hh:mm xxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx xxxxxxxx
= xxxxxxxx.nnn xxxxxxxx mm/dd/yyyy hh:mm xxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx xxxxxxxx
. . . . .
. . . . .
. . . . .

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

```
yy/mm/dd hh:mm ----- PJS(tm) - LIST JOB REQUESTS ----- LINE nnnn OF nnnn
COMMAND ==> _____ SCROLL ==> xxxx

S Request-ID      Member      JCL Data Set Name      Instdata
= xxxxxxxx.nnn xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx xxxxxxxx
= xxxxxxxx.nnn xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx xxxxxxxx
= xxxxxxxx.nnn xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx xxxxxxxx
= xxxxxxxx.nnn xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx xxxxxxxx
= xxxxxxxx.nnn xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx xxxxxxxx
= xxxxxxxx.nnn xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx xxxxxxxx
= xxxxxxxx.nnn 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, J - Browse Saved JCL
```

This panel can be scrolled up and down to display more job requests. Use **PF7** and **PF8** keys to scroll the data up and down.

On an 80-column display, part of the panel extends beyond the right border. Use the **PF10** and **PF11** keys to scroll the data left and right. As you scroll the **S** and **Request-ID** fields keep their position:

You can enter this panel from the **Job Request Menu Panel**

The following command can be entered on the **COMMAND** line:

ADD This will display the **Add Job Request** panel. This panel is described in Subsection 5.2.3.

The following line commands can be selected by typing the appropriate letter in the **S** field for one of the displayed job requests:

C is short for **Copy**. This option will display the **Add Job Requests** panel. This panel is described in Subsection 5.2.2. The values of a copied job request serve as defaults for the new job request to be added. The input fields on the **Add Job Request Panel** will contain the values assigned to the copied job request as will the fields on the **Specify Job Frequency Panel** and the **Specify Job Request Events Panel**

D is short for **Delete**. This option will display the **Delete Job Request** panel for the selected job request. This panel is described in Subsection 5.2.7.

M is short for **Modify**. This option will display the **Modify Job Request** panel for the selected job request. This panel is described in Subsection 5.2.4.

S is short for **Select or diSplay**. This option will display the **Display Job Request** panel for the selected job request. This panel is described in Subsection 5.2.8.

J is short for **JCL**. This option will display the **Browse Saved JCL** panel for the selected job request. This panel is described in Subsection 5.2.10.

You can exit this panel and return to the **PJS Job Request Menu** panel by pressing **PF3**.

The following fields are displayed for each job request:

Request-ID	is the unique ID for the job request. For more information please refer to Section 3.2.
Status	is the status of the job request. For more information please refer to Section 3.7.
Next Date/Time	is the date and time on which PJS is to submit the job. The format of this information field is <i>mm/dd/yyyy hh:mm</i> .
Frequency	is how often PJS is to submit the job request. Valid values are:
ONCE	means the job is submitted one time.
<i>nnn</i> MINUTES	means the job is to be submitted every <i>nnn</i> minutes.
<i>nnn</i> HOURS	means the job is to be submitted every <i>nnn</i> hours.

nnn **DAYS** means the job is to be submitted every *nnn* days.

nnn **WEEKS** means the job is to be submitted every *nnn* weeks.

nn **MONTHS** means the job is to be submitted every *nn* months.

nn **YEARS** means the job is to be submitted every *nn* years.

day[, day]... means the job is to be submitted the displayed weekdays.
Valid values for *day* are:

SU Sunday.
MO Monday.
TU Tuesday.
WE Wednesday.
TH Thursday.
FR Friday.
SA Saturday.

EOM-*nn* means the job is to be submitted *nn* days before the end of each month.

CAL-*calendar-ID* means that the indicated PJS calendar was specified for the job request.

CAL-MULTI means that multiple calendars were specified. Up to three calendars can be specified. The job request must be displayed individually to see all of the specified calendars.

Events

tells you how many job request events are posted. Valid values are:

NONE means no events were specified for the job request.

pe **OF** *ne* means that some, but not all, of the job request events are posted.
The number of posted events is displayed against the total number of events:

pe is the number of posted events.
ne is the total number of events.

POSTED means that all job request events are posted.

Member is the name of the member that contains the job.

JCL Data Set Name is the name of the JCL data set.

Instdata is short for **Installation Data**. This information field can be used to display site-specific data. For example, the PJS Installation Data Format Exit can use this field to display site-specific information.

5.2.3 Add Job Request Panel

Use the PJS Add Job Request Panel to create and place a job request on the PJS Request Queue:

```

yy/mm/dd hh:mm ----- PJS(tm) - ADD JOB REQUEST -----
COMMAND ==> _____

Request-ID ==> xxxxxxxx.nnn          Status          ==> xxxxxxxx

Enabled      ==> xxx                (Yes or No)

JCL Data Set ==> xxxxxxxx.xxxxxxxxx.xxxxxxxxx.xxxxxxxxx.xxxxxxxxx
Member       ==> xxxxxxxx
Save         ==> xxx                (Yes or No)

Start Date   ==> mm/dd/yyyy        (for one-time job request)
Start Time   ==> hh:mm

Commands ==> FREQ - Specify Job Frequency, EVENT - Specify Events,
              END - Complete Add, CANCEL - Cancel Add

```

You can enter this panel from the **Job Request Menu Panel** or the **List Job Requests Panel**

The following informational fields are displayed:

Request-ID is the unique ID for the job request to be added. However, before the add is complete the request number has not yet been assigned. As a result, on this panel, the request-number portion of the request-ID will always be **000**. For more information please refer to Section 3.2.

Status is the status of the job request. For more information please refer to Section 3.7.

The following data entry fields are available:

Enabled tells PJS whether you want to submit the job.

YES puts the job request in WAIT status. When the job request is placed on the PJS Request Queue, PJS can submit the job for execution.

NO puts the job request in DISABLED status. The job request is placed on the PJS Job Request Queue, but PJS will not submit the job for execution. You can use this value to enter the information for a job request, but temporarily prevent job submission.

JCL Data Set is the name of the JCL data set. This is always a fully qualified data set name. Do not use quotes. The name you specify will be saved in your ISPF profile and used as a default in the future.

Member is the name of the member of the JCL Data Set that contains the job. If the JCL data set is a partitioned data set (PDS), you must specify a valid value in this field.

Save tells PJS whether to place a copy of the job JCL on the PJS JCL Spool. If you place job JCL on the PJS JCL Spool, PJS can provide protection from accidental modification or unauthorized access. However, if your site uses a security package such as RACF, your job JCL may already be protected from intrusive access. Valid values are:

NO tells PJS not to place a copy of the job JCL on the PJS JCL Spool. When the job is submitted, the JCL in the JCL data set at the time of submission will be executed.

YES tells PJS to place a copy of the job JCL on the PJS JCL Spool. When the job is submitted, the JCL on the PJS JCL Spool will be executed.

The default for PJS JCL Spool use was determined when PJS was installed. Your site may require you to specify either **NO** or **YES**, or you may be able to specify either value. For more information, please contact your Site Administrator or the person who installed PJS.

Start Date tells PJS which date you want to submit the job for execution. The default is the current system date, which is usually today's date. You can specify an **absolute** date in the *mm/dd/yyyy* format or a **relative** date in the **+ddd* format. For more information on date specifications, please refer to Section 3.6.

Start Time tells PJS what time you want to submit the job for execution. There is no default, you must specify a valid time. You can specify an **absolute** time in the *hh:mm[AM|PM]* format or a **relative** time in the **+{hh:mm|hh:mm}* format. For more information on time specifications, please refer to Section 3.5.

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

FREQ displays the **Specify Job Frequency** panel. This panel enables you to tell PJS how often to submit the job. If you do not use this panel, the job is only submitted once. For more information on the **Specify Job Frequency** panel, please refer to Subsection 5.2.5.

EVENT displays the **Specify Job Request Events** panel. This panel enables you to any events that must be posted before the job is submitted. For more information on the **Specify Job Request Events** panel, please refer to Subsection 5.2.6.

END completes that add and exits the panel after processing the values you entered. Use this command to create a PJS job request that will be placed on the PJS Request Queue. You can also press the **PF3** key to perform this function.

CANCEL exits the panel without processing any of the values you may have entered. Use this command if you decide *not* to create a PJS job request.

5.2.4 Modify Job Request Panel

Use the PJS Modify Job Request Panel to modify a job request on the PJS Request Queue:

```
yy/mm/dd hh:mm ----- PJS(tm) - MODIFY JOB REQUEST -----  
COMMAND ==> =====  
  
Request-ID      ==> xxxxxxxx.nnn          Status           ==> xxxxxxxx  
  
Enabled         ==> xxx                  (Yes or No)  
  
JCL Data Set   ==> xxxxxxxx.xxxxxxxxx.xxxxxxxxx.xxxxxxxxx.xxxxxxxxx  
Member        ==> xxxxxxxx  
Save          ==> xxxxxxx              (Yes, No, or Refresh)  
  
Start Date     ==> mm/dd/yyyy          (for one-time job request)  
Start Time     ==> hh:mm  
  
               Installation Data Heading xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx  
Inst Data      ==> xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx  
  
Commands ==> FREQ - Specify Job Frequency, EVENT - Specify Events,  
             JCL - Browse Saved JCL,  
             END - Complete Modify, CANCEL - Cancel Modify
```

The following informational fields are displayed:

Request-ID is the unique ID for the job request to be modified. For more information please refer to Section 3.2.

Status is the status of the job request. For more information please refer to Section 3.7.

Inst Data is an information field that can be used to display site-specific data.

The following data entry fields are available:

Enabled tells PJS whether you want to submit the job.

YES puts the job request in WAIT status. When the job request is placed on the PJS Request Queue, PJS can submit the job for execution.

NO puts the job request in DISABLED status. The job request is placed on the PJS Job Request Queue, but PJS will not submit the job for execution. You can use this value to enter the information for a job request, but temporarily prevent job submission.

JCL Data Set is the name of the JCL data set. This is always a fully qualified data set name. Do not use quotes.

Member	is the name of the member of the JCL Data Set that contains the job. If the JCL data set is a partitioned data set (PDS), you must specify a valid value in this field.
Save	tells PJS whether to place a copy of the job JCL on the PJS JCL Spool. If you place job JCL on the PJS JCL Spool, PJS can provide protection from accidental modification or unauthorized access. However, if your site uses a security package such as RACF, your job JCL may already be protected from intrusive access. Valid values are: NO tells PJS not to place a copy of the job JCL on the PJS JCL Spool. When the job is submitted, the JCL in the JCL data set at the time of submission will be executed. YES tells PJS to place a copy of the job JCL on the PJS JCL Spool. When the job is submitted, the JCL on the PJS JCL Spool will be executed. REFRESH tells PJS to replace the copy of the job JCL on the PJS JCL Spool. Use this keyword if you have already saved a copy of the job JCL and want to apply any new changes or updates to job JCL. This option is only valid if a copy of the job request's JCL has already been saved in the PJS JCL Spool.
<p>The default for PJS JCL Spool use was determined when PJS was installed. Your site may require you to specify either NO or YES, or you may be able to specify either value. For more information, please contact your Site Administrator or the person who installed PJS.</p>	
Start Date	tells PJS which date you want to submit the job for execution. You can specify an absolute date in the <i>mm/dd/yyyy</i> format or a relative date in the <i>*+ddd</i> format. For more information on date specifications, please refer to Section 3.6.
Start Time	tells PJS what time you want to submit the job for execution. You can specify an absolute time in the <i>hh:mm[AM PM]</i> format or a relative time in the <i>*+{hh:mm hh:mm}</i> format. For more information on time specifications, please refer to Section 3.5.

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

FREQ	displays the Specify Job Frequency panel. This panel enables you to tell PJS how often to submit the job. For more information on the Specify Job Frequency panel, please refer to Subsection 5.2.5.
EVENT	displays the Specify Job Request Events panel. This panel enables you to any events that must be posted before the job is submitted. For more information on the Specify Job Request Events panel, please refer to Subsection 5.2.6.
JCL	displays the Browse Saved JCL panel. This panel enables you to review the JCL that has been saved in the PJS JCL Spool. This command is only valid if a copy of the job request's JCL has already been saved in the PJS JCL Spool. For more information on the Browse Saved JCL panel, please refer to Subsection 5.2.10.

- END** completes that modify and exits the panel after processing the values you entered. You can also press the **PF3** key to perform this function.
- CANCEL** exits the panel without processing any of the values you may have entered. Use this command if you decide *not* to create a PJS job request.

5.2.5 Specify Job Frequency Panel

Use the Specify Job Frequency Panel to tell PJS how often to submit the job:

```

yy/mm/dd hh:mm ----- PJS(tm) - SPECIFY JOB FREQUENCY -----
COMMAND ==> _____

Request-ID   ==> xxxxxxxx.nnn          Status           ==> xxxxxxxx

Start Date   ==> mm/dd/yyyy            End Date         ==> mm/dd/yyyy
Start Time   ==> hh:mm                  End Time         ==> hh:mm

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

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

Commands ==> EVENT - Specify Events, JCL - Browse Saved JCL,
              END - Complete Updates, CANCEL - Cancel Updates

```

You can enter this panel from the **Add Job Request Panel**, the **Modify Job Request Panel**, or the **Specify Job Request Events** panel.

The following informational fields are displayed:

Request-ID is the unique ID for the job request. For more information please refer to Section 3.2.

Status is the status of the job request. For more information please refer to Section 3.7.

The following data entry fields are available:

Start Date tells PJS which date you want to submit the job for execution. You can specify an **absolute** date in the *mm/dd/yyyy* format or a **relative** date in the **+ddd* format. For more information on date specifications, please refer to Section 3.6.

Start Time tells PJS what time you want to submit the job for execution. You can specify an **absolute** time in the *hh:mm[AM|PM]* format or a **relative** time in the **+{hh:mm|hh|mm}* format. For more information on time specifications, please refer to Section 3.5.

End Date tells PJS the date on which it will stop submitting the job for execution. You can specify an **absolute** date in the *mm/dd/yyyy* format or a **relative** date in the **+ddd* format. For more information on date specifications, please refer to Section 3.6.

End Time tells PJS at what time it is to stop submitting the job for execution. If you plan to use the end time parameter, you must also have values specified in the **End Date** input field. You can specify an **absolute** time in the *hh:mm[AM|PM]* format or a **relative** time in the **+{hh:mm|hh:mm}* format. For more information on time specifications, please refer to Section 3.5.

Window Time specifies a window of time during which PJS can submit your job. A job can be submitted late because PJS was down or because events were not posted. While the time window is open, PJS may resume operation or the events may be posted. If your job isn't submitted while the time window is open, the action specified by the **Window Optn** field is taken. The **Window Optn** parameter is discussed below.

The **Window Time** field specifies a time relative to the value specified in the **Start Time** field to determine when the window closes. For example, if submission time is 10:00 and you want to open a time window for two hours, you can enter a relative time value of 2:00. The time window will close at 12:00.

To specify the window time, use the *hh:mm* format, where:

hh is the number of hours. Valid values are **00** through 99.

mm is the number of minutes. Valid values are **00** through 99.

If you don't want a time window for the job request, you can overtype the value in the **Window Time** field with blank characters. A job request without a window time can be delayed indefinitely, and will still be submitted as soon as all its events have been posted and the PJS System Task is up.

Window Opt tells PJS what to do if the job cannot be submitted within the time window specified by the **Window Time** field. You must specify a value for **Window Time** if you plan to use this parameter.

You can specify one of the following:

DISABLE places the job request in DISABLED status. The job request remains on the PJS Request Queue, but it is inactive. PJS cannot submit the job.

ERROR places the job request in ERROR status. The job request remains on the PJS Request Queue until it is modified or deleted. This is the default.

SKIP causes PJS to behave as if the job was submitted. PJS calculates a new run date and time, and resets all job request events. When the new start time value is reached, PJS will attempt to submit the job.

To activate the job request, you can specify **Yes** in the **Enabled** field on the **Modify Job Request Panel**. For more information on the **Modify Job Request Panel**, please refer to Subsection 5.2.4.

Periodic means that you want PJS to submit the job for execution at regular intervals.

This field consists of two subfields, *num* and *units*, where:

num is the number that specifies how many *units* elapse between each job submission. If you specify *unit* as **YR** for year or **MO** for month, valid values range from **1** through **99**. For any other *unit*, valid values range from **1** through **999**.

unit is a measure of time multiplied by *num* to determine how much time elapses between job submissions. When you specify a *unit*, you can use the displayed abbreviations, the full name of the *unit*, or you can abbreviate the full name by specifying the first two or three characters in the *unit* name. You can specify one of the following valid values:

YR means that the job is to be submitted for execution every *num* years.

MO means that the job is to be submitted for execution every *num* months.

Caution: If the number of days in a month is less than the value of **Start Date**, PJS does not submit your job during that month.

For example, if the start date for a monthly job is January 31, PJS assumes you want to submit the job on the 31st of each month and that every month has 31 days. To calculate the next run date, it adds 31 days to January 31: the next run date is February 31st. The job is actually submitted 31 days from January 31. In most years, the next actual run date is March 3, so the job is not submitted during the month of February. The job is next submitted on March 31, so the monthly job is submitted twice in March.

You can avoid this problem by specifying a start date that begins no later than the 28th day of the month. For example, if you specify **1 MO** and a start date of January 16, PJS will submit the job on the 16th of each month.

WK means that the job is to be submitted for execution every *num* weeks.

DAY means that the job is to be submitted for execution every *num* days.

HR means that the job is to be submitted for execution every *num* hours.

MIN means that the job is to be submitted for execution every *num* minutes.

If you specify a value for **Periodic**, you cannot also specify a value in any of the other **Frequency** fields.

Day of Week means that the specified job is submitted for execution every week on each specified days. Select days by typing a non-blank character in the input field that precedes the appropriate day. To remove a selection, overtype the character in the input field with a blank character. You can select from one through seven days.

If you specify a value for **Day of Week**, you cannot also specify a value in any of the other **Frequency** fields.

End of Month means that the job is to be submitted *nn* days before the **end of each month**, where *nn* is a value from **0** through **27**.

For example, if you specify **0**, the job is submitted each month on the last day of the month, for example, May 31 or June 30. If you specify **3**, the job is submitted three days before the end of each month, for example, May 28 or June 27.

If you specify a value for **End of Month**, you cannot also specify a value in any of the other **Frequency** fields.

Calendar(s) means that PJS will submit the job for execution on the date or dates specified in a calendar. This enables you to specify an arbitrary set of submission dates. Before you can use this keyword, you must have created at least one calendar. For information on how to create a PJS calendar, please refer to Section 5.3.

You can specify up to three calendars. However, if you specify more than one calendar, PJS only submits the job on the dates that are specified on every calendar.

Specify each *calendar-ID* with the [*owner-ID*.]*cal-name* format, where:

owner-ID specifies who "owns" the calendar.

cal-name is the name of the calendar.

For more information on how to specify a *calendar-ID*, please refer to Section 3.3.

If you specify a value for **Calendar(s)**, you cannot also specify a value in any of the other **Frequency** fields.

Once is the default for the **Frequency** field. If you do not enter a value in any other **Frequency** field, the job will be submitted for execution once only.

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

EVENT displays the **Specify Job Request Events** panel. This panel enables you to any events that must be posted before the job is submitted. For more information on the **Specify Job Request Events** panel, please refer to Subsection 5.2.6.

JCL displays the **Browse Saved JCL** panel. This panel enables you to review the JCL that has been saved in the PJS JCL Spool. This command is only valid if a copy of the job request's JCL has already been saved in the PJS JCL Spool. For more information on the **Browse Saved JCL** panel, please refer to Subsection 5.2.10.

END completes the information entered on this panel and returns to the **Add Job Request** panel or the **Modify Job Request** panel. You can also press the **PF3** key to perform this function.

CANCEL cancels all updates (or the add) for the current job request. Use this command if you decide *not* to create or modify the PJS job request. PJS will return to the **Job Request Menu** panel or the **List Job Requests** panel.

5.2.6 Specify Job Request Events Panel

Use the Specify Job Request Events Panel to specify events for a job request:

```

yy/mm/dd hh:mm ---- PJS(tm) - SPECIFY JOB REQUEST EVENTS - LINE nnnn OF nnnn
COMMAND ==> _____ SCROLL ==> xxxx

Request-ID ==> xxxxxxxx.nnn Status ==> xxxxxxxx

S      Event-ID      Prepost  Date/Time Posted
=      xxxxxxxx.xxxxxxx xxx    mm/dd/yyyy hh:mm
=      xxxxxxxx.xxxxxxx xxx
=      xxxxxxxx.xxxxxxx xxx
=      xxxxxxxx.xxxxxxx xxx    mm/dd/yyyy hh:mm
=      ===== =====
=      ===== =====
=      ===== =====
=      ===== =====
=      ===== =====
=      ===== =====
=      ===== =====
=      ===== =====
=      ===== =====
=      ===== =====

Commands ==> FREQ - Specify Job Frequency, JCL - Browse Saved JCL,
              END - Complete Updates, CANCEL - Cancel Updates
Line Commands ==> D - Delete Event, P - Post Job Event, R - Reset Job Event

```

You can enter this panel from the **Add Job Request** panel, the **Modify Job Request** panel, or the **Specify Job Request Events** panel.

The following informational fields are displayed:

Request-ID is the unique ID for the job request. For more information please refer to Section 3.2.

Status is the status of the job request. For more information please refer to Section 3.7.

The remainder of the screen displays the job request events that are defined for this job request. The list of job request events can be scrolled up and down to display more job request events. Use **PF7** and **PF8** keys to scroll the data up and down.

The following fields are displayed for each job request event:

Event-ID is the event-ID for the job request event. After all existing job request events are displayed, the remaining lines on the screen are formatted with blank entry fields. To add a new job request event, simply enter an Event-ID in a blank Event-ID field. Specify each *event-ID* with the *[owner-ID.]event-name* format, where:

owner-ID specifies who "owns" the event.
event-name is the name of the event.

For more information on how to specify an *event-ID*, please refer to Section 3.4.

The Event-ID field for existing job request events is protected from updates. To change the Event-ID you must delete the existing job request event and add a new one with the new Event-ID.

Prepost tells PJS whether the job request event can be posted prior to the Next Run Date and Time for the job request. This field can also be changed by overtyping the existing value. Valid values are:

NO means that the event will not be posted in the job request record if the event is posted before the job is scheduled for its next submission.

YES means that the job request event will be posted in the job request record whenever the event is posted. PJS doesn't care when the event is posted.

Date/Time Posted is the date and time on which PJS posted the event for this job request. The format of this information field is *mm/dd/yyyy hh:mm:ss*. If the job request event is not posted, this field is blank.

When an existing job request event appears on a line, the following line commands can be selected by typing the appropriate letter in the **S** field for that job request event:

D is short for **Delete**. This option will delete the job request event.

P is short for **Post**. This option will post the job request event. This will post the event for this job request only. Other job requests are not affected.

R is short for **Reset**. This option will reset the job request event. This will reset the event for this job request only. Other job requests are not affected.

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

FREQ displays the **Specify Job Frequency** panel. This panel enables you to tell PJS how often to submit the job. For more information on the **Specify Job Frequency** panel, please refer to Subsection 5.2.5.

- JCL** displays the **Browse Saved JCL** panel. This panel enables you to review the JCL that has been saved in the PJS JCL Spool. This command is only valid if a copy of the job request's JCL has already been saved in the PJS JCL Spool. For more information on the **Browse Saved JCL** panel, please refer to Subsection 5.2.10.
- END** completes the information entered on this panel and returns to the **Add Job Request** panel or the **Modify Job Request** panel. You can also press the **PF3** key to perform this function.
- CANCEL** cancels all updates (or the add) for the current job request. Use this command if you decide *not* to create or modify the PJS job request. PJS will return to the **Job Request Menu** panel or the **List Job Requests** panel.

5.2.7 Delete Job Request Panel

Use the Delete Job Request Panel to remove a job request from the PJS Request Queue:

```
yy/mm/dd hh:mm ----- PJS(tm) - DELETE JOB REQUEST -----  
COMMAND ==>  
  
Request-ID      ==> xxxxxxxx.nnn           Status       ==> xxxxxxxx  
  
JCL Data Set    ==> xxxxxxxx.xxxxxxxxxx.xxxxxxxxxx.xxxxxxxxxx.xxxxxxxxxx  
Member         ==> xxxxxxxx  
Saved          ==> xxx  
  
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       ==> xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx  
Events          ==> xxxxxxxx  
  
                Installation Data Heading xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx  
Inst Data       ==> xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx  
  
Enter 'YES' and press enter to confirm delete ==> ____  
  
Commands ==> EVENT - Display Events, JCL - Browse Saved JCL,  
             END   - Complete Delete, CANCEL - Cancel Delete
```

You can enter this panel from the **Job Request Menu** panel or the **List Job Requests** panel.

The following informational fields are displayed:

Request-ID is the Request-ID for the job request to be deleted. For more information please refer to Section 3.2.

Status is the status of the job request. For more information please refer to Section 3.7.

JCL Data Set is the name of the JCL data set.

Member is the name of the member of the JCL Data Set that contains the job.

Save indicates if a copy of the job JCL was saved in the PJS JCL Spool. Valid values are:

NO means that the job JCL has not been copied to the PJS JCL Spool. When the job is submitted, the JCL in the JCL data set at the time of submission will be executed.

YES means that the job JCL has been copied to the PJS JCL Spool. When the job is submitted, the JCL on the PJS JCL Spool will be executed.

Next Run D/T is short for **Next Run Date and Time**, that is, the next date and time on which PJS will submit the job. The format of this information field is *mm/dd/yyyy hh:mm*. If PJS is not going to submit or resubmit the job, this field is blank.

Last Run D/T is short for **Last Run Date and Time**, that is, the last date and time on which PJS submitted the job. The format of this information field is *mm/dd/yyyy hh:mm:ss*. If PJS never submitted the job, this field is blank.

Start D/T is short for **Starting Run Date and Time**. This is the first date and time on which PJS was to submit the job. The format of this information field is *mm/dd/yyyy hh:mm*.

End D/T is short for **End Run Date and Time**. This is the last date and time on which PJS is to submit the job. The format of this information field is *mm/dd/yyyy hh:mm*. If the last date and time was never specified, this field is blank.

Frequency is how often PJS is to submit the job request. Valid values are:

ONCE means the job is submitted one time.

nnn **MINUTES** means the job is to be submitted every *nnn* minutes.

nnn **HOURS** means the job is to be submitted every *nnn* hours.

nnn **DAYS** means the job is to be submitted every *nnn* days.

nnn **WEEKS** means the job is to be submitted every *nnn* weeks.

nn **MONTHS** means the job is to be submitted every *nn* months.

nn **YEARS** means the job is to be submitted every *nn* years.

day[, day]... means the job is to be submitted the displayed weekdays. Valid values for *day* are:

SU Sunday.
MO Monday.
TU Tuesday.
WE Wednesday.
TH Thursday.
FR Friday.
SA Saturday.

EOM-*nn* means the job is to be submitted *nn* days before the end of each month.

CAL-*cal-ID* means that the indicated PJS calendar was specified for the job request. Multiple calendars may appear.

Events tells you how many job request events are posted. Valid values are:

NONE means no events were specified for the job request.

pe **OF** *ne* means that some, but not all, of the job request events are posted. The number of posted events is displayed against the total number of events:

pe is the number of posted events.
ne is the total number of events.

POSTED means that all job request events are posted.

Inst Data is an information field that can be used to display site-specific data.

The following data entry field is available:

Enter 'YES'... is the delete confirmation field. You must enter **YES** in this field, then press the **ENTER** key, then enter the **END** command (or press **PF3**), to complete the delete.

If you decide not to delete the job request, enter the **CANCEL** command.

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

- EVENT** displays the **Display Job Request Events** panel. For more information on the **Display Job Request Events** panel, please refer to Subsection 5.2.9.
- JCL** displays the **Browse Saved JCL** panel. This panel enables you to review the JCL that has been saved in the PJS JCL Spool. This command is only valid if a copy of the job request's JCL has already been saved in the PJS JCL Spool. For more information on the **Browse Saved JCL** panel, please refer to Subsection 5.2.10.
- END** completes the delete and returns to the **Job Request Menu** panel or the **List Job Requests** panel. You must enter 'YES' in the delete confirmation field. You can also press the **PF3** key to perform this function.
- CANCEL** exits the panel and returns to the **Job Request Menu** panel or the **List Job Requests** panel without performing any processing. Use this command if you decide *not* to delete the job request.

5.2.8 Display Job Request Panel

Use the Display Job Request Panel to display one job request on the PJS Request Queue:

```
yy/mm/dd hh:mm ----- PJS(tm) - DISPLAY JOB REQUEST -----  
COMMAND ==>  
  
Request-ID      ==> xxxxxxxx.nnn          Status           ==> xxxxxxxx  
  
JCL Data Set    ==> xxxxxxxx.xxxxxxxxxx.xxxxxxxxxx.xxxxxxxxxx.xxxxxxxxxx  
Member         ==> xxxxxxxx  
Save           ==> xxx  
  
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       ==> xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx  
  
Events          ==> xxxxxxxx  
  
                Installation Data Heading xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx  
Inst Data       ==> xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx  
  
Commands ==> EVENT - Display Events, JCL - Browse Saved JCL
```

You can enter this panel from the **Job Request Menu** panel or the **List Job Requests** panel.

The following informational fields are displayed:

Request-ID is the Request-ID for the job request. For more information please refer to Section 3.2.

Status is the status of the job request. For more information please refer to Section 3.7.

JCL Data Set is the name of the JCL data set.

Member is the name of the member of the JCL Data Set that contains the job.

Save indicates if a copy of the job JCL was saved in the PJS JCL Spool. Valid values are:

NO means that the job JCL has not been copied to the PJS JCL Spool. When the job is submitted, the JCL in the JCL data set at the time of submission will be executed.

YES means that the job JCL has been copied to the PJS JCL Spool. When the job is submitted, the JCL on the PJS JCL Spool will be executed.

Next Run D/T is short for **Next Run Date and Time**, that is, the next date and time on which PJS will submit the job. The format of this information field is *mm/dd/yyyy hh:mm*. If PJS is not going to submit or resubmit the job, this field is blank.

Last Run D/T is short for **Last Run Date and Time**, that is, the last date and time on which PJS submitted the job. The format of this information field is *mm/dd/yyyy hh:mm:ss*. If PJS never submitted the job, this field is blank.

Start D/T is short for **Starting Run Date and Time**. This is the first date and time on which PJS was to submit the job. The format of this information field is *mm/dd/yyyy hh:mm*.

End D/T is short for **End Run Date and Time**. This is the last date and time on which PJS is to submit the job. The format of this information field is *mm/dd/yyyy hh:mm*. If the last date and time was never specified, this field is blank.

Frequency is how often PJS is to submit the job request. Valid values are:

ONCE means the job is submitted one time.

nnn **MINUTES** means the job is to be submitted every *nnn* minutes.

nnn **HOURS** means the job is to be submitted every *nnn* hours.

nnn **DAYS** means the job is to be submitted every *nnn* days.

nnn **WEEKS** means the job is to be submitted every *nnn* weeks.

nn **MONTHS** means the job is to be submitted every *nn* months.

nn **YEARS** means the job is to be submitted every *nn* years.

day[, day]... means the job is to be submitted the displayed weekdays. Valid values for *day* are:

SU Sunday.
MO Monday.
TU Tuesday.
WE Wednesday.
TH Thursday.
FR Friday.
SA Saturday.

EOM-*nn* means the job is to be submitted *nn* days before the end of each month.

CAL-*cal-ID* means that the indicated PJS calendar was specified for the job request. Multiple calendars may appear.

- Events** tells you how many job request events are posted. Valid values are:
- NONE** means no events were specified for the job request.
- pe* **OF** *ne* means that some, but not all, of the job request events are posted. The number of posted events is displayed against the total number of events:
- pe* is the number of posted events.
ne is the total number of events.
- POSTED** means that all job request events are posted.
- Inst Data** is an information field that can be used to display site-specific data.

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

- EVENT** displays the **Display Job Request Events** panel. For more information on the **Display Job Request Events** panel, please refer to Subsection 5.2.9.
- JCL** displays the **Browse Saved JCL** panel. This panel enables you to review the JCL that has been saved in the PJS JCL Spool. This command is only valid if a copy of the job request's JCL has already been saved in the PJS JCL Spool. For more information on the **Browse Saved JCL** panel, please refer to Subsection 5.2.10.
- END** returns to the **Job Request Menu** panel or the **List Job Requests** panel. You can also press the **PF3** key to perform this function.

5.2.9 Display Job Request Events Panel

Use the Display Job Request Events Panel to see which events have been specified for a job request:

```
yy/mm/dd hh:mm ---- PJS(tm) - DISPLAY JOB REQUEST EVENTS - LINE nnnn OF nnnn
COMMAND ==> _____ SCROLL ==> xxxx

Request-ID ==> xxxxxxxx.nnn Status ==> xxxxxxxx

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

Commands ==> JCL - Browse Saved JCL
```

You can enter this panel from the **Display Job Request** panel or the **Delete Job Request** panel.

The following informational fields are displayed:

Request-ID is the unique ID for the job request. For more information please refer to Section 3.2.

Status is the status of the job request. For more information please refer to Section 3.7.

The remainder of the screen displays the job request events that are defined for this job request. The list of job request events can be scrolled up and down to display more job request events. Use **PF7** and **PF8** keys to scroll the data up and down.

The following fields are displayed for each job request event:

Event-ID is the event-ID for the job request event. For more information please refer to Section 3.4.

Prepost	tells PJS whether the job request event can be posted prior to the Next Run Date and Time for the job request. Valid values are: NO means that the event will not be posted in the job request record if the event is posted before the job is scheduled for its next submission. YES means that the job request event will be posted in the job request record whenever the event is posted. PJS doesn't care when the event is posted.
Date/Time Posted	is the date and time on which PJS posted the event for this job request. The format of this information field is <i>mm/dd/yyyy hh:mm:ss</i> . If the job request event is not posted, this field is blank.

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

JCL	displays the Browse Saved JCL panel. This panel enables you to review the JCL that has been saved in the PJS JCL Spool. This command is only valid if a copy of the job request's JCL has already been saved in the PJS JCL Spool. For more information on the Browse Saved JCL panel, please refer to Subsection 5.2.10.
END	returns to the Display Job Request panel or the Delete Job Request panel. You can also press the PF3 key to perform this function.

5.2.10 Browse Saved JCL Panel

Use the Browse Saved JCL Panel to review JCL saved on the PJS JCL Spool:

```

BROWSE      xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxx Line nnnnnnnnn Col nnn nnn
COMMAND ==> _____ Scroll ==> xxx
***** Top of Data *****
//      JCL
//      STATEMENTS
//      .
//      .
//      .
***** Bottom of Data *****

```

You can enter this panel from the **List Job Request** panel, the **Modify Job Request** panel, the **Delete Job Request** panel, the **Display Job Request** panel, the **Specify Job Frequency** panel, the **Specify Job Request Events** panel, or the **Display Job Request Events** panel.

The JCL is displayed using the ISPF browse facility. For detailed instructions on how to use ISPF Browse, consult your ISPF documentation. You cannot edit the displayed JCL.

5.3 PJS Calendar Panel System

The following diagram provides an overview of the PJS calendar panel system:

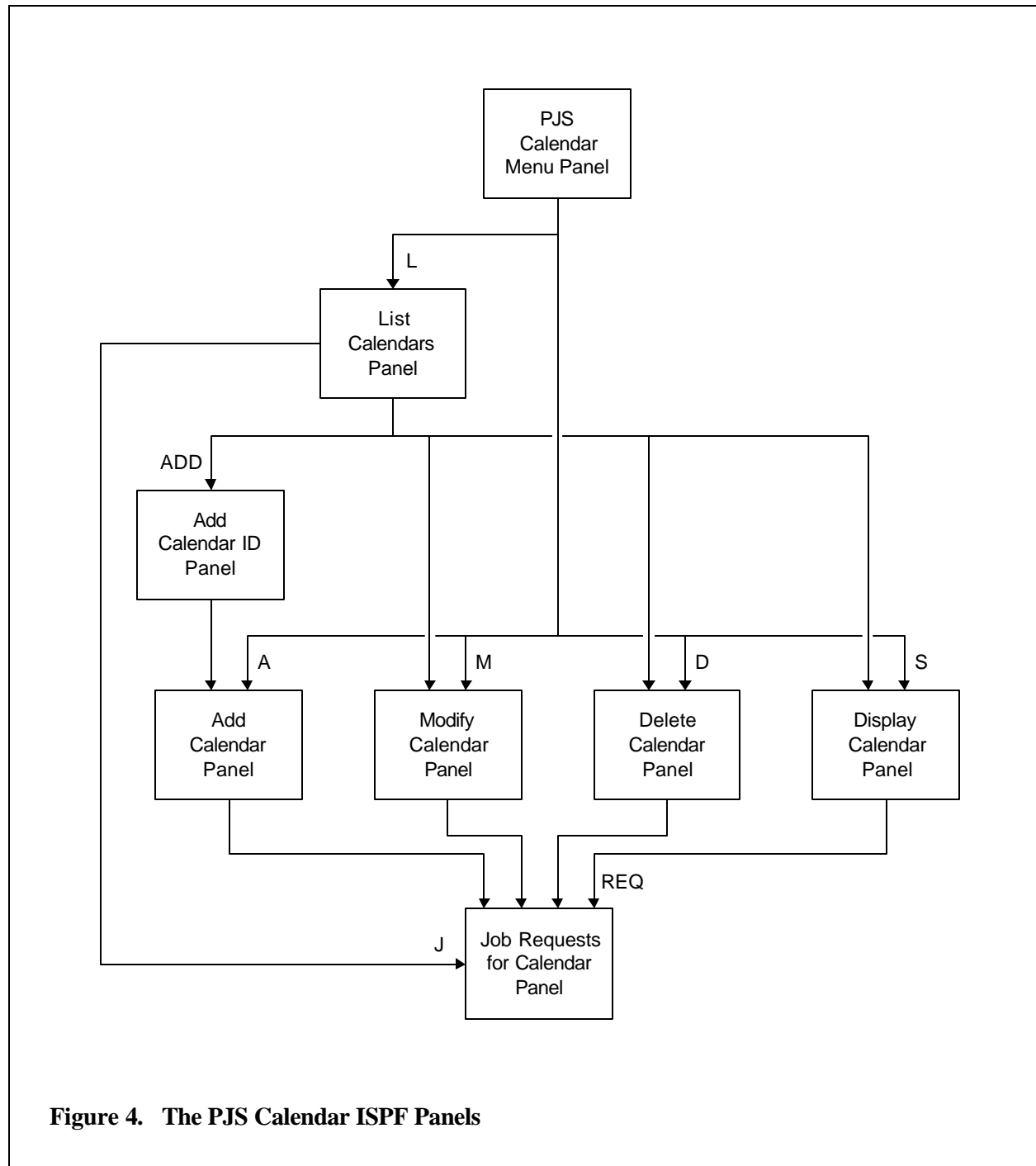


Figure 4. The PJS Calendar ISPF Panels

The **Calendar Menu Panel** is the gateway to the PJS calendar panel system. From here, you can list, add, modify, delete, or display calendars. This panel is discussed in Subsection 5.3.1.

The **List Calendars Panel** lists calendars and their characteristics. You can also add, modify, delete, or display calendars. Additionally, you can display a list of which job requests have specified a given calendar. If you plan to work with more than one calendar during a session, it's a good idea to use this panel. The **List Calendars Panel** is discussed in Subsection 5.3.2.

The **Add Calendar ID Panel** is displayed when you add or copy a panel from the **List Calendars Panel**. It enables you to specify a calendar-ID for the new calendar. The **Add Calendar ID Panel** is discussed in Subsection 5.3.3.

The **Add Calendar Panel** enables you to create a PJS calendar. To build a calendar, you must select at least one date. The dates you select are the dates on which you want PJS to submit a job. You must create a calendar before you can specify one in a job request. This panel is discussed in Subsection 5.3.4.

The **Modify Calendar Panel** enables you to add or delete selected dates on a PJS calendar. This panel is discussed in Subsection 5.3.5.

The **Delete Calendar Panel** enables you to delete a calendar. If you want to remove the calendar specification from a job request, but keep the calendar, please use the **Modify Job Request Panel**, which is discussed in Subsection 5.2.4. If, however, you want to delete the calendar, the **Delete Calendar Panel** is discussed in Subsection 5.3.6.

The **Display Calendar Panel** displays the characteristics of one PJS calendar. This panel is discussed in Subsection 5.3.7.

The **Job Requests for Calendar Panel** displays the characteristics of a job request in which a given PJS calendar was specified. This panel is discussed in Subsection 5.3.8.

5.3.1 Calendar Menu Panel

Use the Calendar Menu Panel as a gateway to the PJS calendar panel system:

```
yy/mm/dd hh:mm ----- PJS(tm) - 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      ==> xxxxxxxx
Calendar Name ==> xxxxxxxx      (Required for options A, D, M, and S)

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

You can enter this panel from the **PJS Main Menu Panel**

The following menu options can be selected by typing the appropriate letter on the **OPTIONS** command line:

- L** is short for **List**. This option will display the **List Calendars** panel. This panel is described in Subsection 5.3.2. When using this option the PJS owner-ID to be listed should be entered in the **Owner-ID** input field.
- A** is short for **Add**. This option will display the **Add Calendar** panel. This panel is described in Subsection 5.3.4. When using this option the PJS owner-ID for the calendar to be added should be entered in the **Owner-ID** input field.
- D** is short for **Delete**. This option will display the **Delete Calendar** panel. This panel is described in Subsection 5.3.6. When using this option the PJS calendar-ID for the calendar to be deleted should be entered in the **Owner-ID** and **Calendar Name** input fields.
- M** is short for **Modify**. This option will display the **Modify Calendar** panel. This panel is described in Subsection 5.3.5. When using this option the PJS calendar-ID for the calendar to be modified should be entered in the **Owner-ID** and **Calendar Name** input fields.
- S** is short for **Select or diSplay**. This option will display the **Display Calendar** panel. This panel is described in Subsection 5.3.7. When using this option the PJS calendar-ID for the calendar to be displayed should be entered in the **Owner-ID** and **Calendar Name** input fields.

You can exit this panel and return to the **PJS Main Menu** panel by pressing **PF3**.

The following input fields are available:

- Owner-ID** specifies who "owns" the calendar you are adding, deleting, modifying, listing, or displaying. You must be authorized to access calendars for an owner-ID other than your own. For more information on the *owner-ID* specification, please refer to Section 3.1.
- Calendar Name** is the name of the calendar. If you plan to add, delete, modify, or display a calendar, you must enter a value in this input field. For more information on this specification, please refer to Section 3.3.

5.3.2 List Calendars Panel

Use the List Calendars Panel to display a set of calendars for a specified *owner-ID*:

```
yy/mm/dd hh:mm ----- PJS(tm) - LIST CALENDARS ----- LINE nnnn OF nnnn
COMMAND ==> _____ SCROLL ==> xxxx

S      Calendar-ID      Next Date
=      xxxxxxxx.xxxxxxxx mm/dd/yyyy
=      xxxxxxxx.xxxxxxxx mm/dd/yyyy
=      xxxxxxxx.xxxxxxxx mm/dd/yyyy
=      xxxxxxxx.xxxxxxxx mm/dd/yyyy
.      .                .
.      .                .
.      .                .

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
```

You can enter this panel from the **Calendar Menu Panel**.

This panel can be scrolled up and down to display more calendars. Use **PF7** and **PF8** keys to scroll the data up and down.

The following command can be entered on the **COMMAND** line:

ADD This will display the **Add Calendar-ID** panel. This panel is described in Subsection 5.3.3.

The following line commands can be selected by typing the appropriate letter in the **S** field for one of the displayed calendars:

C is short for **Copy**. This option will display the **Add Calendar-ID** panel. This panel is described in Subsection 5.3.3. After you enter a new calendar name on this panel, the **Add Calendar** panel will be displayed. This panel is described in Subsection 5.3.4. The dates of the copied calendar serve as defaults for the new calendar to be added. The new calendar that appears on the **Add Calendar** panel will have the same dates selected as in the copied calendar.

D is short for **Delete**. This option will display the **Delete Calendar** panel for the selected calendar. This panel is described in Subsection 5.3.6.

M is short for **Modify**. This option will display the **Modify Calendar** panel for the selected calendar. This panel is described in Subsection 5.3.5.

- S** is short for **Select or diSplay**. This option will display the **Display Calendar** panel for the selected calendar. This panel is described in Subsection 5.3.7.
- J** is short for **Job Requests**. This option will display the **List Job Requests for Calendar** panel for the selected calendar. This panel is described in Subsection 5.3.8.

You can exit this panel and return to the **PJS Calendar Menu** panel by pressing **PF3**.

The following fields are displayed for each calendar:

Calendar-ID is the calendar-ID for the calendar. For more information please refer to Section 3.3.

Next Date is the next date selected on the calendar. The format of this information field is *mm/dd/yyyy*. If this field is blank, either no dates are selected, so the calendar is empty; or the last selected date on the calendar has passed.

5.3.3 Add Calendar ID Panel

Use the Add Calendar ID Panel to specify a calendar-ID for a new calendar:

```

yy/mm/dd hh:mm ----- PJS(tm) - ADD CALENDAR ID -----
COMMAND ==> _____

New Calendar-ID:
  Owner-ID      ==> XXXXXXXX
  Calendar Name ==> XXXXXXXX

Enter New Calendar-ID, then press ENTER.

Commands ==> CANCEL - Cancel Add

```

You can enter this panel from the **List Calendars** panel.

The following data entry fields are available:

Owner-ID specifies who "owns" the calendar you are adding or copying. For more information on how to specify an *owner-ID*, please refer to Section 3.1.

Calendar Name is the name of the calendar. For more information on how to specify this value, please refer to Section 3.3.

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

CANCEL cancels the add for the calendar. Use this command if you decide *not* to create the PJS calendar. PJS will return to the **List Calendars** panel.

5.3.4 Add Calendar Panel

Use the Add Calendar Panel to create a PJS calendar:

```
yy/mm/dd hh:mm ----- PJS(tm) - ADD CALENDAR -----
COMMAND ==> _____

Calendar-ID ==> .XXXXXXXX.XXXXXXXXXX

      Month XXXXXXXX                Year YYYY

      Sun   Mon   Tue   Wed   Thu   Fri   Sat
      x 06 x 07 x 08 x 09 x 10 x 11 x 12
      x 13 x 14 x 15 x 16 x 17 x 18 x 19
      x 20 x 21 x 22 x 23 x 24 x 25 x 26
      x 27 x 28 x 29 x 30 x 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
```

You can enter this panel from the **Calendar Menu** panel, or the **List Calendars** panel through the **Add Calendar ID** panel.

The following informational fields are displayed:

Calendar-ID is the Calendar-ID for the calendar to be added. For more information please refer to Section 3.3.

The calendar area of this panel shows the calendar, one month at a time, in the typical calendar format. The month to be displayed can be selected by either typing the month and year to be displayed into the provided input fields, or by “scrolling” through the calendar using the normal ISPF scroll commands (or PF keys.) To scroll to an earlier month, use the **UP** command (or **PF7**). To scroll to a later month use the **DOWN** command (or **PF8**).

The following data entry fields are available:

Month is the name of the month to be displayed in the calendar area. Only the first three characters entered are checked. Use the following three-letter values:

JAN	APR	JUL	OCT
FEB	MAY	AUG	NOV
MAR	JUN	SEP	DEC

Year is the year. You can enter a value that ranges from the current year through **2099**.

Dates are dates selected by the calendar. To select a date, place any non-blank character, in the input field that precedes the date. If you want to deselect a date, overwrite the character with a space. When PJS displays the selected dates, it will use an asterisk ('*') and highlight the date.

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

REQ displays the **Job Requests for Calendar** panel. This panel will display all of the job requests that specify this calendar. However, since you are adding the calendar, there will likely be no job requests to list. For more information on the **Job Requests for Calendar** panel, please refer to Subsection 5.3.8.

END completes that add and exits the panel after processing the values you entered. Use this command to create the PJS calendar. You can also press the **PF3** key to perform this function.

CANCEL exits the panel and returns to the **Calendar Menu** panel or the **List Calendars** panel without performing any processing. Use this command if you decide *not* to add the calendar.

5.3.5 Modify Calendar Panel

Use the Modify Calendar Panel to modify an existing PJS calendar:

```

yy/mm/dd hh:mm ----- PJS(tm) - MODIFY CALENDAR -----
COMMAND ==> _____

Calendar-ID ==> .XXXXXXXX.XXXXXXXXXX

      Month XXXXXXXX                Year YYYY

      Sun   Mon   Tue   Wed   Thu   Fri   Sat
      x 06 x 07 x 01 x 02 x 03 x 04 x 05
      x 13 x 14 x 08 x 09 x 10 x 11 x 12
      x 20 x 21 x 15 x 16 x 17 x 18 x 19
      x 27 x 28 x 22 x 23 x 24 x 25 x 26
      x 29 x 30 x 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 Modify, CANCEL - Cancel Modify

```

You can enter this panel from the **Calendar Menu** panel, or the **List Calendars** panel.

The following informational fields are displayed:

Calendar-ID is the Calendar-ID for the calendar to be modified. For more information please refer to Section 3.3.

The calendar area of this panel shows the calendar, one month at a time, in the typical calendar format. The month to be displayed can be selected by either typing the month and year to be displayed into the provided input fields, or by “scrolling” through the calendar using the normal ISPF scroll commands (or PF keys.) To scroll to an earlier month, use the **UP** command (or **PF7**). To scroll to a later month use the **DOWN** command (or **PF8**).

The following data entry fields are available:

Month is the name of the month to be displayed in the calendar area. Only the first three characters entered are checked. Use the following three-letter values:

JAN	APR	JUL	OCT
FEB	MAY	AUG	NOV
MAR	JUN	SEP	DEC

- Year** is the year. You can enter a value that ranges from the current year through **2099**.
- Dates** are dates selected by the calendar. To select a date, place any non-blank character, in the input field that precedes the date. If you want to deselect a date, overwrite the character with a space. When PJS displays the selected dates, it will use an asterisk ('*') and highlight the date.

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

- REQ** displays the **Job Requests for Calendar** panel. This panel will display all of the job requests that specify this calendar. For more information on the **Job Requests for Calendar** panel, please refer to Subsection 5.3.8.
- END** completes that modify and exits the panel after processing the values you entered. You can also press the **PF3** key to perform this function.
- CANCEL** exits the panel and returns to the **Calendar Menu** panel or the **List Calendars** panel without performing any processing. Use this command if you decide *not* to modify the calendar.

5.3.6 Delete Calendar Panel

Use the Delete Calendar Panel to delete an existing PJS calendar:

```
yy/mm/dd hh:mm ----- PJS(tm) - DELETE CALENDAR -----
COMMAND ==> _____

Calendar-ID ==> xxxxxxxx.xxxxxxxx

      Month xxxxxxxx                Year yyyy

      Sun   Mon   Tue   Wed   Thu   Fri   Sat
      * 06   07   * 01   02   03   * 04   05
      * 13   14   * 08   09   10   * 11   * 12
      * 20   * 21   * 15   * 16   * 17   18   19
      27   28   29   * 22   23   24   * 25   * 26
      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 'YES' and press enter to confirm delete ==> xxx

Commands ==> REQ - Display Job Requests for Calendar,
              END - Complete Delete, CANCEL - Cancel Delete
```

You can enter this panel from the **Calendar Menu** panel, or the **List Calendars** panel.

The following informational fields are displayed:

Calendar-ID is the Calendar-ID for the calendar to be deleted. For more information please refer to Section 3.3.

The calendar area of this panel shows the calendar, one month at a time, in the typical calendar format. The month to be displayed can be selected by either typing the month and year to be displayed into the provided input fields, or by “scrolling” through the calendar using the normal ISPF scroll commands (or PF keys.) To scroll to an earlier month, use the **UP** command (or **PF7**). To scroll to a later month use the **DOWN** command (or **PF8**).

The dates selected by the calendar are indicated by an asterisk (*) preceding the date, and the date is highlighted.

The following data entry fields are available:

Month is the name of the month to be displayed in the calendar area. Only the first three characters entered are checked. Use the following three-letter values:

JAN	APR	JUL	OCT
FEB	MAY	AUG	NOV
MAR	JUN	SEP	DEC

Year is the year. You can enter a value that ranges from the current year through **2099**.

Enter 'YES'... is the delete confirmation field. You must enter **YES** in this field, then press the **ENTER** key, then enter the **END** command (or press **PF3**), to complete the delete.

If you decide not to delete the calendar, enter the **CANCEL** command.

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

REQ displays the **Job Requests for Calendar** panel. This panel will display all of the job requests that specify this calendar. For more information on the **Job Requests for Calendar** panel, please refer to Subsection 5.3.8.

END completes that delete and returns to the **Calendar Menu** panel or the **List Calendars** panel. You must enter '**YES**' in the delete confirmation field. You can also press the **PF3** key to perform this function.

CANCEL exits the panel and returns to the **Calendar Menu** panel or the **List Calendars** panel without performing any processing. Use this command if you decide *not* to delete the calendar.

5.3.7 Display Calendar Panel

Use the Display Calendar Panel to display one PJS calendar:

```
yy/mm/dd hh:mm ----- PJS(tm) - DISPLAY CALENDAR -----
COMMAND ==> _____

Calendar-ID ==> .XXXXXXXX.XXXXXXXXXX

      Month XXXXXXXX                Year YYYY

      Sun   Mon   Tue   Wed   Thu   Fri   Sat
      * 06    07    * 01    02    03    * 04    05
      * 13    14    08    * 09    10    11    * 12
      * 20    21    15    * 16    * 17    18    19
      27    28    29    * 30    * 31    24    * 25    * 26

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
```

You can enter this panel from the **Calendar Menu** panel, or the **List Calendars** panel.

The following informational fields are displayed:

Calendar-ID is the Calendar-ID for the calendar to be deleted. For more information please refer to Section 3.3.

The calendar area of this panel shows the calendar, one month at a time, in the typical calendar format. The month to be displayed can be selected by either typing the month and year to be displayed into the provided input fields, or by “scrolling” through the calendar using the normal ISPF scroll commands (or PF keys.) To scroll to an earlier month, use the **UP** command (or **PF7**). To scroll to a later month use the **DOWN** command (or **PF8**).

The dates selected by the calendar are indicated by an asterisk (*) preceding the date, and the date is highlighted.

The following data entry fields are available:

Month is the name of the month to be displayed in the calendar area. Only the first three characters entered are checked. Use the following three-letter values:

JAN	APR	JUL	OCT
FEB	MAY	AUG	NOV
MAR	JUN	SEP	DEC

Year is the year. You can enter a value that ranges from the current year through **2099**.

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

REQ displays the **Job Requests for Calendar** panel. This panel will display all of the job requests that specify this calendar. For more information on the **Job Requests for Calendar** panel, please refer to Subsection 5.3.8.

END returns to the **Calendar Menu** panel or the **List Calendars** panel. You can also press the **PF3** key to perform this function.

5.3.8 Job Requests for Calendar Panel

Use the Job Requests for Calendar Panel to display job requests with the calendar specified:

yy/mm/dd hh:mm --- PJS(tm) - JOB REQUESTS FOR CALENDAR --- LINE nnnn OF nnnn					
COMMAND ==>				SCROLL ==> xxxx	
Request-ID	Status	Next Date/Time	Frequency	Events	Member
xxxxxxxx.nnn	xxxxxxxx	mm/dd/yyyy hh:mm	xxxxxxxxxxxxxxxxxxxxxxxxxxxx	xxxxxxxx	xxxxxxxx
xxxxxxxx.nnn	xxxxxxxx	mm/dd/yyyy hh:mm	nnn YR	NONE	xxxxxxxx
xxxxxxxx.nnn	xxxxxxxx	mm/dd/yyyy hh:mm	SU,MO,TU,WE,TH,FR,SA	nn OF nn	xxxxxxxx
xxxxxxxx.nnn	xxxxxxxx	mm/dd/yyyy hh:mm	EOM-nn	POSTED	xxxxxxxx
xxxxxxxx.nnn	xxxxxxxx	mm/dd/yyyy hh:mm	CAL-xxxxxxxx.xxxxxxxxx	xxxxxxxx	xxxxxxxx
xxxxxxxx.nnn	xxxxxxxx	mm/dd/yyyy hh:mm	xxxxxxxxxxxxxxxxxxxxxxxxxxxx	xxxxxxxx	xxxxxxxx
xxxxxxxx.nnn	xxxxxxxx	mm/dd/yyyy hh:mm	xxxxxxxxxxxxxxxxxxxxxxxxxxxx	xxxxxxxx	xxxxxxxx
.
.
.

yy/mm/dd hh:mm --- PJS(tm) - JOB REQUESTS FOR CALENDAR --- LINE nnnn OF nnnn					
COMMAND ==>				SCROLL ==> xxxx	
Request-ID	Member	JCL Data Set Name	Instdata		
xxxxxxxx.nnn	xxxxxxxx	xxxxxxxx.xxxxxxxxx.xxxxxxxxx.xxxxxxxxx.xxxxxxxxx	xxxxxxxx		
xxxxxxxx.nnn	xxxxxxxx	xxxxxxxx.xxxxxxxxx.xxxxxxxxx.xxxxxxxxx.xxxxxxxxx	xxxxxxxx		
xxxxxxxx.nnn	xxxxxxxx	xxxxxxxx.xxxxxxxxx.xxxxxxxxx.xxxxxxxxx.xxxxxxxxx	xxxxxxxx		
xxxxxxxx.nnn	xxxxxxxx	xxxxxxxx.xxxxxxxxx.xxxxxxxxx.xxxxxxxxx.xxxxxxxxx	xxxxxxxx		
xxxxxxxx.nnn	xxxxxxxx	xxxxxxxx.xxxxxxxxx.xxxxxxxxx.xxxxxxxxx.xxxxxxxxx	xxxxxxxx		
xxxxxxxx.nnn	xxxxxxxx	xxxxxxxx.xxxxxxxxx.xxxxxxxxx.xxxxxxxxx.xxxxxxxxx	xxxxxxxx		
xxxxxxxx.nnn	xxxxxxxx	xxxxxxxx.xxxxxxxxx.xxxxxxxxx.xxxxxxxxx.xxxxxxxxx	xxxxxxxx		
.
.
.

You can enter this panel from the **List Calendars** panel, **Add Calendar** panel, **Display Calendar** panel, **Modify Calendar** panel, or **Delete Calendar** panel.

This panel can be scrolled up and down to display more job requests. Use **PF7** and **PF8** keys to scroll the data up and down. On an 80-column display, part of the panel extends beyond the right border. Use the **PF10** and **PF11** keys to scroll the data left and right. As you scroll the **Request-ID** field keeps its position:

You can exit this panel by pressing **PF3**.

The following fields are displayed for each job request:

Request-ID	is the unique ID for the job request. For more information please refer to Section 3.2.																																				
Status	is the status of the job request. For more information please refer to Section 3.7.																																				
Next Date/Time	is the date and time on which PJS is to submit the job. The format of this information field is <i>mm/dd/yyyy hh:mm</i> .																																				
Frequency	is how often PJS is to submit the job request. Valid values are: <table data-bbox="472 590 1409 1728"> <tr> <td>ONCE</td><td>means the job is submitted one time.</td></tr> <tr> <td><i>nnn</i> MINUTES</td><td>means the job is to be submitted every <i>nnn</i> minutes.</td></tr> <tr> <td><i>nnn</i> HOURS</td><td>means the job is to be submitted every <i>nnn</i> hours.</td></tr> <tr> <td><i>nnn</i> DAYS</td><td>means the job is to be submitted every <i>nnn</i> days.</td></tr> <tr> <td><i>nnn</i> WEEKS</td><td>means the job is to be submitted every <i>nnn</i> weeks.</td></tr> <tr> <td><i>nn</i> MONTHS</td><td>means the job is to be submitted every <i>nn</i> months.</td></tr> <tr> <td><i>nn</i> YEARS</td><td>means the job is to be submitted every <i>nn</i> years.</td></tr> <tr> <td><i>day[, day]...</i></td><td>means the job is to be submitted the displayed weekdays. Valid values for <i>day</i> are: <table data-bbox="760 1161 984 1396"> <tr><td>SU</td><td>Sunday.</td></tr> <tr><td>MO</td><td>Monday.</td></tr> <tr><td>TU</td><td>Tuesday.</td></tr> <tr><td>WE</td><td>Wednesday.</td></tr> <tr><td>TH</td><td>Thursday.</td></tr> <tr><td>FR</td><td>Friday.</td></tr> <tr><td>SA</td><td>Saturday.</td></tr> </table> </td></tr> <tr> <td>EOM-<i>nn</i></td><td>means the job is to be submitted <i>nn</i> days before the end of each month.</td></tr> <tr> <td>CAL-<i>calendar-ID</i></td><td>means that the indicated PJS calendar was specified for the job request.</td></tr> <tr> <td>CAL-MULTI</td><td>means that multiple calendars were specified. Up to three calendars can be specified. The job request must be displayed individually to see all of the specified calendars.</td></tr> </table>	ONCE	means the job is submitted one time.	<i>nnn</i> MINUTES	means the job is to be submitted every <i>nnn</i> minutes.	<i>nnn</i> HOURS	means the job is to be submitted every <i>nnn</i> hours.	<i>nnn</i> DAYS	means the job is to be submitted every <i>nnn</i> days.	<i>nnn</i> WEEKS	means the job is to be submitted every <i>nnn</i> weeks.	<i>nn</i> MONTHS	means the job is to be submitted every <i>nn</i> months.	<i>nn</i> YEARS	means the job is to be submitted every <i>nn</i> years.	<i>day[, day]...</i>	means the job is to be submitted the displayed weekdays. Valid values for <i>day</i> are: <table data-bbox="760 1161 984 1396"> <tr><td>SU</td><td>Sunday.</td></tr> <tr><td>MO</td><td>Monday.</td></tr> <tr><td>TU</td><td>Tuesday.</td></tr> <tr><td>WE</td><td>Wednesday.</td></tr> <tr><td>TH</td><td>Thursday.</td></tr> <tr><td>FR</td><td>Friday.</td></tr> <tr><td>SA</td><td>Saturday.</td></tr> </table>	SU	Sunday.	MO	Monday.	TU	Tuesday.	WE	Wednesday.	TH	Thursday.	FR	Friday.	SA	Saturday.	EOM-<i>nn</i>	means the job is to be submitted <i>nn</i> days before the end of each month.	CAL-<i>calendar-ID</i>	means that the indicated PJS calendar was specified for the job request.	CAL-MULTI	means that multiple calendars were specified. Up to three calendars can be specified. The job request must be displayed individually to see all of the specified calendars.
ONCE	means the job is submitted one time.																																				
<i>nnn</i> MINUTES	means the job is to be submitted every <i>nnn</i> minutes.																																				
<i>nnn</i> HOURS	means the job is to be submitted every <i>nnn</i> hours.																																				
<i>nnn</i> DAYS	means the job is to be submitted every <i>nnn</i> days.																																				
<i>nnn</i> WEEKS	means the job is to be submitted every <i>nnn</i> weeks.																																				
<i>nn</i> MONTHS	means the job is to be submitted every <i>nn</i> months.																																				
<i>nn</i> YEARS	means the job is to be submitted every <i>nn</i> years.																																				
<i>day[, day]...</i>	means the job is to be submitted the displayed weekdays. Valid values for <i>day</i> are: <table data-bbox="760 1161 984 1396"> <tr><td>SU</td><td>Sunday.</td></tr> <tr><td>MO</td><td>Monday.</td></tr> <tr><td>TU</td><td>Tuesday.</td></tr> <tr><td>WE</td><td>Wednesday.</td></tr> <tr><td>TH</td><td>Thursday.</td></tr> <tr><td>FR</td><td>Friday.</td></tr> <tr><td>SA</td><td>Saturday.</td></tr> </table>	SU	Sunday.	MO	Monday.	TU	Tuesday.	WE	Wednesday.	TH	Thursday.	FR	Friday.	SA	Saturday.																						
SU	Sunday.																																				
MO	Monday.																																				
TU	Tuesday.																																				
WE	Wednesday.																																				
TH	Thursday.																																				
FR	Friday.																																				
SA	Saturday.																																				
EOM-<i>nn</i>	means the job is to be submitted <i>nn</i> days before the end of each month.																																				
CAL-<i>calendar-ID</i>	means that the indicated PJS calendar was specified for the job request.																																				
CAL-MULTI	means that multiple calendars were specified. Up to three calendars can be specified. The job request must be displayed individually to see all of the specified calendars.																																				

Events	tells you how many job request events are posted. Valid values are: NONE means no events were specified for the job request. <i>pe</i> OF <i>ne</i> means that some, but not all, of the job request events are posted. The number of posted events is displayed against the total number of events: <i>pe</i> is the number of posted events. <i>ne</i> is the total number of events. POSTED means that all job request events are posted.
Member	is the name of the member that contains the job.
JCL Data Set Name	is the name of the JCL data set.
Instdata	is short for Installation Data . This information field can be used to display site-specific data. For example, the PJS Installation Data Format Exit can use this field to display site-specific information.

5.4 PJS Event Panel System

The following diagram provides an overview of the PJS event panel system:

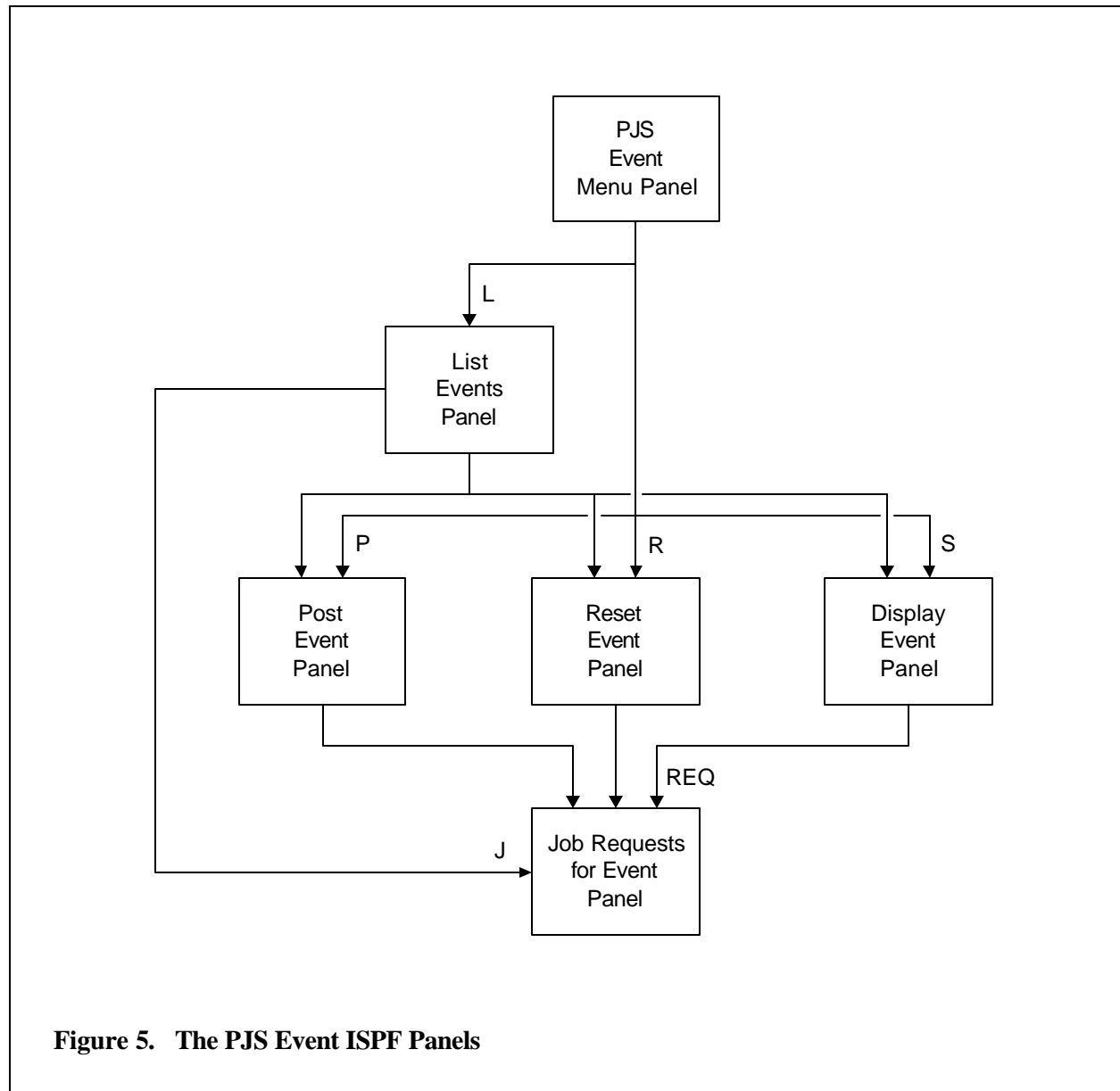


Figure 5. The PJS Event ISPF Panels

The **Event Menu Panel** is the gateway to the PJS event panel system. From here, you can list, post, reset, or display events. This panel is discussed in Subsection 5.4.1.

The **List Events Panel** lists events. You can also post, reset, or display events. Additionally, you can display a list of which job requests have specified a given event. If you plan to work with more than one event during a session, it's a good idea to use this panel. The **List Events Panel** is discussed in Subsection 5.4.2.

The **Post Event Panel** enables you to post an event. Before you can use this panel, an event must be specified in at least one job request. When an event is posted, the event dependency is satisfied in every job request in which the event is specified. If you have an event specified in more than one job request, but want to post the event in one job request, you can use the **Specify Job Request Events Panel**, which is discussed in Subsection 5.2.6. If, however, you plan to post an event in every job request that specifies the event, use this panel. The **Post Event Panel** is discussed in Subsection 5.4.3.

The **Reset Event Panel** enables you to reset an event. Before you can use this panel, an event must be specified in at least one job request. When an event is reset, event posting is removed, so the job request event in every relevant job request must be posted before PJS can submit the job for execution. Events are automatically reset after job submission. If you have an event specified in more than one job request, but want to reset the event in one job request, you can use the **Specify Job Request Events Panel**, which is discussed in Subsection 5.2.6. If, however, you plan to reset an event in every job request that specifies the event, use this panel. The **Reset Event Panel** is discussed in Subsection 5.4.4.

The **Display Event Panel** displays the status of one PJS event. This panel is discussed in Subsection 5.4.5.

The **Job Requests for Event Panel** displays the job requests in which a given event was specified. This panel is discussed in Subsection 5.4.6.

5.4.1 Event Menu Panel

Use the Calendar Menu Panel as a gateway to the PJS calendar panel system:

```
yy/mm/dd hh:mm ----- PJS(tm) - 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      ==> xxxxxxxx
  Event Name    ==> xxxxxxxx      (Required for options A, D, M, and S)

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

You can enter this panel from the **PJS Main Menu Panel**

The following menu options can be selected by typing the appropriate letter on the **OPTIONS** command line:

- L** is short for **List**. This option will display the **List Events** panel. This panel is described in Subsection 5.4.2. When using this option the PJS owner-ID to be listed should be entered in the **Owner-ID** input field.
- P** is short for **Post**. This option will display the **Post Event** panel. This panel is described in Subsection 5.4.3. When using this option the PJS event-ID for the event to be posted should be entered in the **Owner-ID** and **Event Name** input fields.
- R** is short for **Reset**. This option will display the **Reset Event** panel. This panel is described in Subsection 5.4.4. When using this option the PJS event-ID for the event to be reset should be entered in the **Owner-ID** and **Event Name** input fields.
- S** is short for **Select or diSplay**. This option will display the **Display Event** panel. This panel is described in Subsection 5.4.5. When using this option the PJS event-ID for the event to be displayed should be entered in the **Owner-ID** and **Event Name** input fields.

You can exit this panel and return to the **PJS Main Menu** panel by pressing **PF3**.

The following input fields are available:

- Owner-ID** specifies who "owns" the event you are posting, resetting, listing, or displaying. You must be authorized to access events for an owner-ID other than your own. For more information on the *owner-ID* specification, please refer to Section 3.1.
- Event Name** is the name of the event. If you plan to post, reset, or display an event, you must enter a value in this input field. For more information on this specification, please refer to Section 3.4.

5.4.2 List Events Panel

Use the List Events Panel to display a set of events for a specified owner-ID:

```
yy/mm/dd hh:mm ----- PJS(tm) - LIST EVENTS ----- LINE nnnn OF nnnn
COMMAND ==> _____ SCROLL ==> xxxx

^ELSTAREA
^
S      Event-ID          Last Post Date/Time      Status
=  xxxxxxxx.xxxxxxxx    mm/dd/yyyy hh:mm:ss    POST PENDING
=  xxxxxxxx.xxxxxxxx    mm/dd/yyyy hh:mm:ss    RESET PENDING
=  xxxxxxxx.xxxxxxxx    mm/dd/yyyy hh:mm:ss    xxxxxxxxxxxxxx
=  xxxxxxxx.xxxxxxxx    mm/dd/yyyy hh:mm:ss    xxxxxxxxxxxxxx
.      .                  .                  .
.      .                  .                  .
.      .                  .                  .

Line Commands ==> P - Post Event, R - Reset Event, S - Display Event
                  J - List Job Requests for Event
```

You can enter this panel from the **Event Menu Panel**

This panel can be scrolled up and down to display more events. Use **PF7** and **PF8** keys to scroll the data up and down.

The following line commands can be selected by typing the appropriate letter in the **S** field for one of the displayed events:

- P** is short for **Post**. This option will display the **Post Event** panel for the selected event. This panel is described in Subsection 5.4.3.
- R** is short for **Reset**. This option will display the **Reset Event** panel for the selected event. This panel is described in Subsection 5.4.4.
- S** is short for **Select or diSplay**. This option will display the **Display Event** panel for the selected event. This panel is described in Subsection 5.4.5.
- J** is short for **Job Requests**. This option will display the **List Job Requests for Event** panel for the selected event. This panel is described in Subsection 5.4.6.

You can exit this panel and return to the **PJS Event Menu** panel by pressing **PF3**.

The following fields are displayed for each event:

Event-ID is the event-ID for the event. For more information please refer to Section 3.4.

Last Post Date/Time is the date and time PJS last posted the event. The format of this information field is *mm/dd/yyyy hh:mm:ss*. If the event has never been posted, this field is blank.

Status describes the current status of the event. Valid values are:

POST PENDING means that a command to post the event has been received. PJS will wait until the current scan interval elapses, then post the job request events.

RESET PENDING means that a command to reset the event has been received. PJS will wait until the current scan interval elapses, then reset the job request events.

Because PJS usually processes post and reset requests quickly, this field will usually be blank. For more information on how PJS processes events, refer to Section 2.5

5.4.3 Post Event Panel

Use the Post Event Panel to post an event in every relevant job request:

```

yy/mm/dd hh:mm ----- PJS(tm) - POST EVENT -----
COMMAND ==> _____

Event-ID          ==> xxxxxxxx.xxxxxxx

Event Status      ==> xxxxxxxxxxxx

Date/Time Posted  ==> mm/dd/yyyy hh:mm:ss

Enter 'YES' and press enter to confirm post ==> xxx

Commands ==> REQ - Display Job Requests for Event,
              END - Complete Post, CANCEL - Cancel Post

```

You can enter this panel from the **Event Menu** panel, or the **List Events** panel.

The following informational fields are displayed:

Event-ID is the Event-ID for the event to be posted. For more information please refer to Section 3.4.

Date/Time Posted is the date and time PJS last posted the event. The format of this information field is *mm/dd/yyyy hh:mm:ss*. If the event has never been posted, this field is blank.

Status describes the current status of the event. Valid values are:

POST PENDING means that a command to post the event has been received. PJS will wait until the current scan interval elapses, then post the job request events.

RESET PENDING means that a command to reset the event has been received. PJS will wait until the current scan interval elapses, then reset the job request events.

Because PJS usually processes post and reset requests quickly, this field will usually be blank. For more information on how PJS processes events, refer to Section 2.5

The following data entry field is available:

Enter 'YES'... is the post confirmation field. You must enter **YES** in this field, then press the **ENTER** key, then enter the **END** command (or press **PF3**), to complete the post.

If you decide not to post the event, enter the **CANCEL** command.

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

- REQ** displays the **Job Requests for Event** panel. This panel will display all of the job requests that specify this event. For more information on the **Job Requests for Event** panel, please refer to Subsection 5.4.6.
- END** completes that post and returns to the **Event Menu** panel or the **List Events** panel. You must enter '**YES**' in the post confirmation field. You can also press the **PF3** key to perform this function.
- CANCEL** exits the panel and returns to the **Event Menu** panel or the **List Events** panel without performing any processing. Use this command if you decide *not* to post the event.

5.4.4 Reset Event Panel

Use the Reset Event Panel to reset an event in every relevant job request:

```

yy/mm/dd hh:mm ----- PJS(tm) - RESET EVENT -----
COMMAND ==> _____

Event-ID          ==> xxxxxxxx.xxxxxxx
Event Status      ==> xxxxxxxxxxxx
Date/Time Posted  ==> mm/dd/yyyy hh:mm:ss

Enter 'YES' and press enter to confirm reset ==> xxx

Commands ==> REQ - Display Job Requests for Event,
              END - Complete Reset, CANCEL - Cancel Reset

```

You can enter this panel from the **Event Menu** panel, or the **List Events** panel.

The following informational fields are displayed:

Event-ID is the Event-ID for the event to be reset. For more information please refer to Section 3.4.

Date/Time Posted is the date and time PJS last posted the event. The format of this information field is *mm/dd/yyyy hh:mm:ss*. If the event has never been posted, this field is blank.

Status describes the current status of the event. Valid values are:

POST PENDING means that a command to post the event has been received. PJS will wait until the current scan interval elapses, then post the job request events.

RESET PENDING means that a command to reset the event has been received. PJS will wait until the current scan interval elapses, then reset the job request events.

Because PJS usually processes post and reset requests quickly, this field will usually be blank. For more information on how PJS processes events, refer to Section 2.5

The following data entry field is available:

Enter 'YES'... is the reset confirmation field. You must enter **YES** in this field, then press the **ENTER** key, then enter the **END** command (or press **PF3**), to complete the reset.

If you decide not to reset the event, enter the **CANCEL** command.

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

- REQ** displays the **Job Requests for Event** panel. This panel will display all of the job requests that specify this event. For more information on the **Job Requests for Event** panel, please refer to Subsection 5.4.6.
- END** completes that reset and returns to the **Event Menu** panel or the **List Events** panel. You must enter '**YES**' in the reset confirmation field. You can also press the **PF3** key to perform this function.
- CANCEL** exits the panel and returns to the **Event Menu** panel or the **List Events** panel without performing any processing. Use this command if you decide *not* to reset the event.

5.4.5 Display Event Panel

Use the Display Event Panel to display characteristics of one event:

```

yy/mm/dd hh:mm ----- PJS(tm) - DISPLAY EVENT -----
COMMAND ==> _____

Event-ID          ==> xxxxxxxx.xxxxxxx
Event Status      ==> xxxxxxxxxxxxxx
Date/Time Posted ==> mm/dd/yyyy hh:mm:ss

Commands ==> REQ - Display Job Requests for Event

```

You can enter this panel from the **Event Menu** panel, or the **List Events** panel.

The following informational fields are displayed:

Event-ID is the Event-ID for the event. For more information please refer to Section 3.4.

Date/Time Posted is the date and time PJS last posted the event. The format of this information field is *mm/dd/yyyy hh:mm:ss*. If the event has never been posted, this field is blank.

Status describes the current status of the event. Valid values are:

POST PENDING means that a command to post the event has been received. PJS will wait until the current scan interval elapses, then post the job request events.

RESET PENDING means that a command to reset the event has been received. PJS will wait until the current scan interval elapses, then reset the job request events.

Because PJS usually processes post and reset requests quickly, this field will usually be blank. For more information on how PJS processes events, refer to Section 2.5

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

- REQ** displays the **Job Requests for Event** panel. This panel will display all of the job requests that specify this event. For more information on the **Job Requests for Event** panel, please refer to Subsection 5.4.6.
- END** returns to the **Event Menu** panel or the **List Events** panel. You can also press the **PF3** key to perform this function.

5.4.6 Job Requests for Event Panel

Use the Job Requests for Event Panel to display job requests with the event specified:

```
yy/mm/dd hh:mm ----- PJS(tm) - JOB REQUESTS FOR EVENT ----- LINE nnnn OF nnnn
COMMAND ==> _____ SCROLL ==> xxxx

Request-ID   Status   Next Date/Time   Frequency   Events   Member
xxxxxxxx.nnn xxxxxxxx mm/dd/yyyy hh:mm xxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx xxxxxxxx
xxxxxxxx.nnn xxxxxxxx mm/dd/yyyy hh:mm nnn YR          NONE      xxxxxxxx
xxxxxxxx.nnn xxxxxxxx mm/dd/yyyy hh:mm SU,MO,TU,WE,TH,FR,SA nn OF nn xxxxxxxx
xxxxxxxx.nnn xxxxxxxx mm/dd/yyyy hh:mm EOM-nn          POSTED    xxxxxxxx
xxxxxxxx.nnn xxxxxxxx mm/dd/yyyy hh:mm CAL-xxxxxxxx.xxxxxxxxx xxxxxxxx xxxxxxxx
xxxxxxxx.nnn xxxxxxxx mm/dd/yyyy hh:mm xxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx xxxxxxxx
xxxxxxxx.nnn xxxxxxxx mm/dd/yyyy hh:mm xxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx xxxxxxxx
.            .            .            .            .            .
.            .            .            .            .            .
.            .            .            .            .            .
```

```
yy/mm/dd hh:mm ----- PJS(tm) - JOB REQUESTS FOR EVENT ----- LINE nnnn OF nnnn
COMMAND ==> _____ SCROLL ==> xxxx

S Request-ID   Member   JCL Data Set Name   Instdata
= xxxxxxxx.nnn xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx xxxxxxxx
= xxxxxxxx.nnn xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx xxxxxxxx
= xxxxxxxx.nnn xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx xxxxxxxx
= xxxxxxxx.nnn xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx xxxxxxxx
= xxxxxxxx.nnn xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx xxxxxxxx
= xxxxxxxx.nnn xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx xxxxxxxx
= xxxxxxxx.nnn xxxxxxxx xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx xxxxxxxx
. . . . .
. . . . .
. . . . .
```

You can enter this panel from the **List Events Panel**, **Post Event Panel**, **Display Event Panel**, or **Reset Event Panel**.

This panel can be scrolled up and down to display more job requests. Use **PF7** and **PF8** keys to scroll the data up and down. On an 80-column display, part of the panel extends beyond the right border. Use the **PF10** and **PF11** keys to scroll the data left and right. As you scroll the **Request-ID** field keeps its position:

You can exit this panel by pressing **PF3**.

The following fields are displayed for each job request:

Request-ID	is the unique ID for the job request. For more information please refer to Section 3.2.																																				
Status	is the status of the job request. For more information please refer to Section 3.7.																																				
Next Date/Time	is the date and time on which PJS is to submit the job. The format of this information field is <i>mm/dd/yyyy hh:mm</i> .																																				
Frequency	is how often PJS is to submit the job request. Valid values are: <table data-bbox="470 588 1396 1728"> <tr> <td>ONCE</td><td>means the job is submitted one time.</td></tr> <tr> <td><i>nnn</i> MINUTES</td><td>means the job is to be submitted every <i>nnn</i> minutes.</td></tr> <tr> <td><i>nnn</i> HOURS</td><td>means the job is to be submitted every <i>nnn</i> hours.</td></tr> <tr> <td><i>nnn</i> DAYS</td><td>means the job is to be submitted every <i>nnn</i> days.</td></tr> <tr> <td><i>nnn</i> WEEKS</td><td>means the job is to be submitted every <i>nnn</i> weeks.</td></tr> <tr> <td><i>nn</i> MONTHS</td><td>means the job is to be submitted every <i>nn</i> months.</td></tr> <tr> <td><i>nn</i> YEARS</td><td>means the job is to be submitted every <i>nn</i> years.</td></tr> <tr> <td><i>day[, day]...</i></td><td>means the job is to be submitted the displayed weekdays. Valid values for <i>day</i> are: <table data-bbox="747 1155 990 1396"> <tr><td>SU</td><td>Sunday.</td></tr> <tr><td>MO</td><td>Monday.</td></tr> <tr><td>TU</td><td>Tuesday.</td></tr> <tr><td>WE</td><td>Wednesday.</td></tr> <tr><td>TH</td><td>Thursday.</td></tr> <tr><td>FR</td><td>Friday.</td></tr> <tr><td>SA</td><td>Saturday.</td></tr> </table> </td></tr> <tr> <td>EOM-<i>nn</i></td><td>means the job is to be submitted <i>nn</i> days before the end of each month.</td></tr> <tr> <td>CAL-<i>calendar-ID</i></td><td>means that the indicated PJS calendar was specified for the job request.</td></tr> <tr> <td>CAL-MULTI</td><td>means that multiple calendars were specified. Up to three calendars can be specified. The job request must be displayed individually to see all of the specified calendars.</td></tr> </table>	ONCE	means the job is submitted one time.	<i>nnn</i> MINUTES	means the job is to be submitted every <i>nnn</i> minutes.	<i>nnn</i> HOURS	means the job is to be submitted every <i>nnn</i> hours.	<i>nnn</i> DAYS	means the job is to be submitted every <i>nnn</i> days.	<i>nnn</i> WEEKS	means the job is to be submitted every <i>nnn</i> weeks.	<i>nn</i> MONTHS	means the job is to be submitted every <i>nn</i> months.	<i>nn</i> YEARS	means the job is to be submitted every <i>nn</i> years.	<i>day[, day]...</i>	means the job is to be submitted the displayed weekdays. Valid values for <i>day</i> are: <table data-bbox="747 1155 990 1396"> <tr><td>SU</td><td>Sunday.</td></tr> <tr><td>MO</td><td>Monday.</td></tr> <tr><td>TU</td><td>Tuesday.</td></tr> <tr><td>WE</td><td>Wednesday.</td></tr> <tr><td>TH</td><td>Thursday.</td></tr> <tr><td>FR</td><td>Friday.</td></tr> <tr><td>SA</td><td>Saturday.</td></tr> </table>	SU	Sunday.	MO	Monday.	TU	Tuesday.	WE	Wednesday.	TH	Thursday.	FR	Friday.	SA	Saturday.	EOM-<i>nn</i>	means the job is to be submitted <i>nn</i> days before the end of each month.	CAL-<i>calendar-ID</i>	means that the indicated PJS calendar was specified for the job request.	CAL-MULTI	means that multiple calendars were specified. Up to three calendars can be specified. The job request must be displayed individually to see all of the specified calendars.
ONCE	means the job is submitted one time.																																				
<i>nnn</i> MINUTES	means the job is to be submitted every <i>nnn</i> minutes.																																				
<i>nnn</i> HOURS	means the job is to be submitted every <i>nnn</i> hours.																																				
<i>nnn</i> DAYS	means the job is to be submitted every <i>nnn</i> days.																																				
<i>nnn</i> WEEKS	means the job is to be submitted every <i>nnn</i> weeks.																																				
<i>nn</i> MONTHS	means the job is to be submitted every <i>nn</i> months.																																				
<i>nn</i> YEARS	means the job is to be submitted every <i>nn</i> years.																																				
<i>day[, day]...</i>	means the job is to be submitted the displayed weekdays. Valid values for <i>day</i> are: <table data-bbox="747 1155 990 1396"> <tr><td>SU</td><td>Sunday.</td></tr> <tr><td>MO</td><td>Monday.</td></tr> <tr><td>TU</td><td>Tuesday.</td></tr> <tr><td>WE</td><td>Wednesday.</td></tr> <tr><td>TH</td><td>Thursday.</td></tr> <tr><td>FR</td><td>Friday.</td></tr> <tr><td>SA</td><td>Saturday.</td></tr> </table>	SU	Sunday.	MO	Monday.	TU	Tuesday.	WE	Wednesday.	TH	Thursday.	FR	Friday.	SA	Saturday.																						
SU	Sunday.																																				
MO	Monday.																																				
TU	Tuesday.																																				
WE	Wednesday.																																				
TH	Thursday.																																				
FR	Friday.																																				
SA	Saturday.																																				
EOM-<i>nn</i>	means the job is to be submitted <i>nn</i> days before the end of each month.																																				
CAL-<i>calendar-ID</i>	means that the indicated PJS calendar was specified for the job request.																																				
CAL-MULTI	means that multiple calendars were specified. Up to three calendars can be specified. The job request must be displayed individually to see all of the specified calendars.																																				

Events	tells you how many job request events are posted. Valid values are: NONE means no events were specified for the job request. <i>pe</i> OF <i>ne</i> means that some, but not all, of the job request events are posted. The number of posted events is displayed against the total number of events: <i>pe</i> is the number of posted events. <i>ne</i> is the total number of events. POSTED means that all job request events are posted.
Member	is the name of the member that contains the job.
JCL Data Set Name	is the name of the JCL data set.
Instdata	is short for Installation Data . This information field can be used to display site-specific data. For example, the PJS Installation Data Format Exit can use this field to display site-specific information.

6. Examples

To help you feel more comfortable using PJS, this chapter presents several common scheduling problems and solutions:

- Section 6.1, **How to Run a One -Time Job**, shows how to issue a job request for a job that PJS will submit once. Minimal specifications and accepted defaults are discussed.
- Section 6.2, **How to Run a Daily Job**, shows how to issue a job request for a job that PJS will submit every day. Frequency option issues are discussed.
- Section 6.3, **How to Run a Job on the First Tuesday of Each Month**, shows how to specify a PJS calendar and specify the calendar in a job request.
- Section 6.4, **How to Run a Job That Depends on Another Job**, shows how an event can solve certain types of scheduling problems. Job dependency factors and the **PJSPOST JCL** step are discussed.
- Section 6.5, **How to Set Up Several Dependent Jobs**, shows how to use events in an environment where several job dependencies exist.
- Section 6.6, **How to Run a Job Whenever CICS Terminates**, shows how you can use an event on the system level.
- Section 6.7, **How to Run a Weekly Job After a Daily Job**, shows how non-preposted events can be used to solve certain types of scheduling problems. The rest of this chapter assumes that events are preposted.
- Section 6.8, **How to Run a Job After a Manual Operation**, shows how PJS/TSO commands or PJS/ISPF panels can be used to post an event. The rest of this chapter assumes that events are posted by the **PJSPOST JCL** step.
- Section 6.9, **How to Reset an Event**, shows how PJS/TSO commands or PJS/ISPF panels can be used to reset an event.
- Section 6.10, **How to Enable a Failed Job Request**, shows how you can deal with a job request that specifies a job that PJS cannot submit.

If you are new to PJS, it's a good idea to read this chapter from start to finish. Many of the examples assume that you know how to use PJS features illustrated in previous examples. Later, as you become familiar with PJS, you will be able to refer to the specific examples you need.

PJS/TSO commands are used in the examples. All PJS/TSO commands can be replicated on PJS/ISPF panels.

6.1 How to Run a One-Time Job

Suppose you want to submit a job at 3:00 AM tomorrow morning. You can use the following command to create the job request:

```
pjreqadd dataset('qual.tso.data(myjob)') -  
          date(*+1) time(3:00AM)
```

The new job request has the following characteristics:

- The JCL data set name is **QUAL.TSO.DATA(MYJOB)**. The enclosing quotes make this a fully qualified data set name.
- The date of submission is one day added to the current system date. The date is specified as a relative date.
- The time of submission is 3:00 AM, using an absolute time specification. Because a 24-hour clock is assumed, you can specify this value as **3:00** or **3:00AM**: PJS accepts either specification.
- No frequency option was specified, so PJS will submit the job only once.
- Because the job is only submitted once, you don't have to specify an **End Date** or **End Time**.
- Assuming that your site gives you a choice, a copy of job JCL was not saved on the PJS JCL Spool. At submission, the JCL is submitted from the JCL Data Set.
- No **Time Window** was specified, so there is no actual deadline for job submission. If the system is down at time of submission, the job will be submitted after the system comes up and the PJS System Task is started.
- No events or job request events are specified.
- The **ENABLED** default is accepted. The PJS System Task puts the new job request in WAIT status.

6.2 How to Run a Daily Job

Suppose you want to submit a job every day at 6:00 AM. You can use the following command to create the job request:

```
pjreqadd dataset('qual.tso.data(myjob)') save -  
          time(6:00) daily
```

The new job request has the following characteristics:

- The JCL data set name is **QUAL.TSO.DATA(MYJOB)**. The enclosing quotes make this a fully qualified data set name.
- Assuming that your site gives you a choice, the **SAVE** parameter causes PJS create a copy of the job JCL in the JCL data set, and put the copy on the PJS JCL Spool. At submission, the JCL is submitted from the PJS JCL Spool.
- The time of submission is 6:00 AM, using an absolute time specification.
- The **DAILY** frequency option is specified, so PJS will submit the job every day.
- No **Start Date** was specified, so the current system date is assumed. If the system date is today's date, and if you added the job request before 6:00 AM, the job will first be submitted at 6:00 AM today. If it's after 6:00 AM, the job will first be submitted at 6:00 AM tomorrow.
- No **End Date** or **End Time** is specified. If you don't delete the job request, or use **PJREQMOD** to change the job submission characteristics, the job will run every day at 6:00 AM for as long as you have PJS running on the system.
- No **Time Window** was specified, so there is no actual deadline for job submission. If the system is down at time of submission, the job will be submitted after the system comes up and the PJS System Task is started. If the system is down for more than a day, PJS will miss at least one job submission.
- No events or job request events are specified.
- The **ENABLED** default is accepted. The PJS System Task puts the new job request in WAIT status.

6.3 How to Run a Job on the First Tuesday of Each Month

Suppose you want PJS to submit a job on the first Tuesday of every month in 2003 at 6:00 PM. From **MINUTES** through **YEARS**, none of the periodic frequency options give you a way to specify the first Tuesday in every month. However, you can regard the first Tuesday of every month is an arbitrary set of dates. You can select arbitrary sets of dates in a PJS calendar, which must be specified before you can issue a job request that uses the calendar.

You can use the following command to create the PJS calendar:

```
pjcaladd firsttue dates(1/7/03 2/4/03 3/3/92 4/7/03 -  
                        5/5/03 6/2/03 7/7/92 8/4/03 -  
                        9/1/03 10/6/03 11/3/92 12/1/03)
```

The created calendar has the following characteristics:

- The name of the calendar is **FIRSTTUE**. The owner-ID will take the default value, which is your TSO user-ID.
- The **DATES** parameter is inclusive, so the specified dates will be selected on the PJS calendar.

After the calendar has been created, you can specify the calendar as the frequency option in a PJS job request:

```
pjreqadd dataset('qual.tso.data(myjob)') -  
            time(6:00PM) calendar(firsttue)
```

PJS will now submit the job at 6:00 PM on each of the dates specified in the calendar.

6.4 How to Run a Job That Depends on Another Job

Suppose you want PJS to submit two jobs. JOB1 is to be submitted at 6:00 PM today, and JOB2 should be submitted after JOB1 finishes execution. You can approach this problem in a couple of ways:

- You can guess when JOB1 will stop executing, and specify an appropriate **Start Time** value for JOB2. However, you may underestimate the amount of time JOB1 will need. JOB1 submission may be delayed, in which case JOB2 may execute at the same time as JOB1. Making JOB2 job submission time-specific may cause problems.
- You can think of "completion of execution" as an event. An event is not time-specific.

First, make JOB1 post an event by adding the following step to the end of its JCL statements:

```
//JOB1 JOB . . .
// existing JCL statements
//          .
//          .
//          .
//PJSPST EXEC PGM=PJSPST,PARM='USER01.JOB1END'
```

The new job step has the following characteristics:

- The *stepname* is **PJSPST**. Any validly formed name will be accepted.
- The **PGM=PJSPST** statement is required, as is the **PARM** statement.
- The **PARM** statement must specify a full *event-ID*. In this case, your *owner-ID* is **USER01** and the *event-name* is **JOB1END**.

Next, create a job request to run JOB1:

```
pjreqadd dataset('qual.tso.data(job1)') -
         time(18:00)
```

Finally, create a job request to run JOB2:

```
pjreqadd dataset('qual.tso.data(job2)') -
         time(18:00) -
         event(job1end)
```

PJS will submit JOB1 at 6:00 PM. JOB2 is also scheduled to run at 6:00 PM, but it also has an event (JOB1END) that must be posted before it will be submitted. This event is posted by the last step of JOB1. So when JOB1 completes, and the last step is run, the PJSPST batch program will post the JOB1END event. PJS can then submit JOB2.

6.5 How to Set Up Several Dependent Jobs

Suppose you want to submit six jobs, and suppose you have the following constraints:

1. JOB1 does not depend on other jobs, and is to be submitted at 10:00 PM.
2. JOB2 does not depend on other jobs, and is to be submitted at 12:00 AM.
3. JOB3 depends on JOB1, and can be submitted only after JOB1 finishes.
4. JOB4 depends on JOB1 and JOB2, and cannot be submitted before 2:00 AM.
5. JOB5 depends on JOB2, and cannot be submitted before 1:00 AM.
6. JOB6 depends on JOB3, JOB4, and JOB5, and can be submitted as soon as all are finished.

First, make all of the jobs post an event by adding the following step to the end of each job:

```
//JOB $n$  JOB . . .
// existing JCL statements
//
//
//
//
//PJSPOST EXEC PGM=PJSPOST,PARM='USER01.JOB $n$ END'
```

In each job, n is the number of the job. (The actual job names and event names are not significant. These names are used for clarity.)

After all six jobs are modified, issue a job request for each job:

```
pjreqadd dataset('qual.tso.data(job1)') -
         time(10:00PM)

pjreqadd dataset('qual.tso.data(job2)') -
         date(*+1) time(12:00AM)

pjreqadd dataset('qual.tso.data(job3)') -
         time(10:00PM) -
         event(job1end)

pjreqadd dataset('qual.tso.data(job4)') -
         date(*+1) time(2:00AM) -
         event(job1end,job2end)

pjreqadd dataset('qual.tso.data(job5)') -
         date(*+1) time(1:00AM) -
         event(job2end)
```

```
pjreqadd dataset('qual.tso.data(job6)') -  
            date(*+1) time(2:00AM) -  
            event(job3end,job4end,job5end)
```

After creating these job requests:

- JOB1 is scheduled to run at 10:00 PM tonight, with no dependencies.
- JOB2 is scheduled to run at 12:00 AM tomorrow morning (midnight tonight), with no dependencies.
- JOB3 is scheduled to run at 10:00 PM tonight (the same time as JOB1), but will not actually run until JOB1 posts the event JOB1END as it finishes.
- JOB4 is scheduled to run at 2:00 AM tomorrow morning, but will wait until both JOB1 posts the event JOB1END, and JOB2 posts the event JOB2END. If both jobs finish before 2:00 AM, JOB4 will wait and run at its scheduled run time. Otherwise it will wait for the jobs to complete.
- JOB5 is scheduled to run at 1:00 AM tomorrow morning, and will wait until JOB2 posts the event JOB2END. If JOB2 finishes before 1:00 AM, JOB5 will wait and run at its scheduled run time. Otherwise it will wait for JOB2 to complete.
- JOB6 is scheduled to run at 2:00 AM tomorrow morning (the same start time as it's latest predecessor job, JOB4), but will wait until JOB3, JOB4, and JOB5 post the events JOB3END, JOB4END, and JOB5END, respectively. You don't need to specifically wait for JOB1 and JOB2, because JOB4 (as well as JOB3 and JOB5) is already waiting for them. But there would be no harm in specifying those events as well.

6.6 How to Run a Job Whenever CICS Terminates

Suppose you want PJS to submit a job whenever CICS terminates. You can consider CICS termination an event, just you do with a batch job. However, this will probably require the assistance of the CICS system programmer, and may also require the assistance of the PJS administrator.

Add the **PJSPOST** job step to the end of the CICS Task (this will probably have to be done by the CICS system programmer):

```
//CICS PROC . . .  
// existing CICS procedure  
//  
//  
//  
//PJSPOST EXEC PGM=PJSPOST,PARM='CICSA.CICSEND',  
// COND=EVEN
```

Note that the event-ID uses an owner-ID of **CICSA**. (The exact name is not important.) This name is used instead of your own TSO user-ID so that others can also use the same event.

Also, the jobstep includes the **COND=EVEN** JCL parameter to ensure that the post runs even if CICS abends. You may wish to consider if this is appropriate for your application.

Then create the job request:

```
pjreqadd dataset('qual.tso.data(myjob)') -  
         time(*+:1) minute(1) -  
         event(cicsa.cicsend)
```

This new job request is scheduled to run once a minute, starting just one minute from now! But, since the job request also waits on an event, it has the effect of running the job request whenever the event is posted. The time of day is not significant.

When you specify the event-ID, you must include the owner-ID (CICSA), since it is different than your TSO user-ID. To do this you may need your PJS site administrator to change the site security to allow you to do this. Also the site security policy must allow the CICS Task to post the PJS event.

6.7 How to Run a Weekly Job After a Daily Job

Suppose you want to submit two jobs. JOB1 is to be submitted Monday through Friday at 6:00 PM. JOB2 is to be submitted only on Friday and only after JOB1 finishes.

First, make JOB1 post an event by adding the following step to the end of its JCL statements:

```
//JOB1 JOB . . .  
// existing JCL statements  
//  
//  
//  
//PJSPOST EXEC PGM=PJSPOST,PARM='USER01.JOBLEND'
```

Next, issue a job request for JOB1:

```
pjreqadd dataset('qual.tso.data(job1)') -  
          time(6:00PM) wkday(mo,tu,we,th,fr)
```

Finally, issue a job request for JOB2.

```
pjreqadd dataset('qual.tso.data(job2)') -  
          time(6:00PM) wkday(fr) -  
          event(joblend/noprepost)
```

JOB1 is scheduled to run every Monday through Friday at 6:00 PM. JOB2 is scheduled to run every Friday, also at 6:00 PM.

The **EVENT** parameter is used to require that JOB2 wait for JOB1 to finish. However, JOB1 runs and posts the event every Monday, Tuesday, Wednesday, and Thursday, as well as Friday. Without the **/NOPREPOST** option, JOB2 will run on Friday at 6:00 PM, without waiting for JOB1 to finish, because the event was posted earlier in the week. The event normally is not reset in the job request for JOB2 until JOB2 is submitted.

The **/NOPREPOST** option for event posting means that PJS cannot post the job request event before the **Next Run Date and Time**. Since JOB2 is not scheduled to run until Friday at 6:00 PM, the job request event will not be posted when JOB1 runs on Monday through Thursday. However, when JOB1 runs on Friday (which will be after the 6:00 PM scheduled run time for JOB2) the job request event will be posted, and JOB2 will be submitted.

6.8 How to Run a Job After a Manual Operation

Suppose that you, USER01, want to submit a long job after your colleague, USER02, leaves for the day. You can define an event that USER02 can post before leaving.

Assuming you are authorized, you can use the following command to create the job request:

```
pjreqadd dataset('qual.tso.data(myjob)') -  
          time(17:00) -  
          event(user02.imgone)
```

Then, before leaving, USER02 can use the following command to post the event:

```
pjevpost imgone
```

The event will be posted and the job will be submitted at 5:00 PM. If it is already past 5:00 PM the job will be submitted (almost) immediately.

6.9 How to Reset an Event

Suppose that you, USER01, want to submit a long job after your colleague, USER02, leaves for the day. You can use the procedure detailed in Section 6.8.

First, you can issue the following job request:

```
pjreqadd dataset('qual.tso.data(myjob)') -  
          time(17:00) -  
          event(user02.imgone)
```

Then, USER02 can post the event:

```
pjevpost imgone
```

Suppose that USER02 returns to the office before the 5:00 PM runtime for the job. The condition for job submission was USER02's absence. USER02, realizing this, can issue the following command:

```
pjevrset imgone
```

The event is reset, and posting is removed from the event and related job request events. If this command is issued before PJS submits the job, the job is not submitted.

6.10 How to Enable a Failed Job Request

Suppose you issue two job requests, and make JOB2 dependent on JOB1. Now, suppose that the job JCL for JOB2 was accidentally deleted after the job requests were added but before the jobs were submitted.

When JOB1 completes, the **USER01.JOB1END** event is posted. The PJS System task then tries to submit JOB2, but fails. The PJS System Task sends an error message to the user.

Assuming that the *request-ID* is USER01.002, you can use the following command to check job request status:

```
pjreqlst id(2) detail
```

This command generates the following display:

Request-ID	Status	Next Run		Window		Last Submit	
		Date	Time	Time	Option	Date	Time
USER01.002	ERROR	09/10/1991	18:00			09/10/91	8:31:57
JCL Data Set Name						Save	
QUAL.TSO.DATA(JOB2)						NOSAVE	
		Frequency		Start		End	
		Date	Time	Date	Time	Date	Time
		ONCE		09/10/1991	18:00		
		Pre-Post		Posted			
		Date	Time	Date	Time		
		Event-ID					
		USER01.JOB1END	YES	09/10/1991	18:31:56		

You see that the job request was placed in **ERROR** status. The error messages and a little bit of exploration reveal the cause of submission failure. To fix the job request, you can:

- Put JCL in the JCL data set.
- Issue the **PJREQMOD** command and specify a new JCL data set in the **JCLDSN** parameter.

After you fix the problem, you need to enable the job request:

```
premed 2 enable
```

The job request status will change to **WAIT**. In this case, if nothing else is changed, the job will be submitted (almost) immediately, since the **Next Run Date and Time** is in the past and all job request events are still posted. If this is not appropriate then the **STARTDATE** and **STARTTIME** parameters should be used to change the **Next Run Date and Time**, or the **RESETEVENTS** parameter should be used to reset the job request events before enabling the job request.

Appendix A. Summary of Changes

A.1 Changes for PJS® Release 2.1.4

- The CPU authorization checking has been removed.
- Commands to display copyright and license information have been added to the ISPF menu.

A.2 Changes for PJS® Release 2.1.3

- The default century for 2-digit years is changed from 19xx to 20xx.
- PJS Documentation is now available in PDF (Adobe Acrobat Reader) format.

A.3 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 an installation option.
- 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. Secondly, 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.