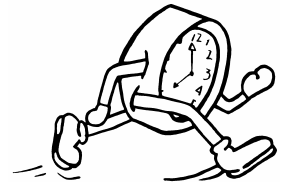


PJS^â

Personal Job Scheduler



Installation Guide

Release 2.1

Related PJS Technical Documentation

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

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	v
Notational Symbols	vi
Acknowledgments	vii
PJS Software Copyright and License	vii
Comments and Suggestions	vii
 1. Introduction	 1
 2. How To Install PJS	 3
2.1 Task 1: Obtain the PJS Installation Package	4
2.2 Task 2: Extract the PJS Installation Package PDS	5
2.3 Task 3: Review the \$COPYRT, \$LICENSE, and \$README Files	7
2.4 Task 4: Extract the PJS Installation Files	7
2.5 Task 5: Extract the PJS Installation Files	8
2.6 Task 6: Allocate the PJS Data Sets	9
2.7 Task 7: Install the PJS Software	11
2.7.1 Installation with SMP/E.....	11
2.7.1.1 SMP/E Subtask 1: Prepare the SMP/E Zones	11
2.7.1.2 SMP/E Subtask 2: Receive, Apply, Accept PJS Function SYSMOD	11
2.7.1.3 SMP/E Subtask 3: Receive and Apply User Exit USERMODs	11
2.7.1.4 SMP/E Subtask 4: Receive and Apply the PJSOPT USERMOD	12
2.7.2 Installation without SMP/E.....	13
2.7.2.1 Non-SMP/E Subtask 1: Copy PJS from the PJS Installation Files	13
2.7.2.2 Non-SMP/E Subtask 2: Link-Edit the PJS Software	13
2.7.2.3 Non-SMP/E Subtask 3: Assemble and Link-Edit User Exits (Optional)	13
2.7.2.4 Non-SMP/E Subtask 4: Assemble and Link-Edit PJSOPT	13
2.8 Task 8: Add PJS to the Link List or the LPA (Optional)	14
2.9 Task 9: Authorize the PJS Load Libraries	14
2.10 Task 10: Initialize the PJS Request Queue	14
2.11 Task 11: Allocate the PJS JCL Spool (Optional)	15
2.12 Task 12: Create the PJS System Task Procedure	15
2.13 Task 13: Set Up the PJS/TSO Interface	15
2.14 Task 14: Set Up the PJS/ISPF Interface (Optional)	16
2.15 Task 15: Start the PJS System Task	16
 3. The PJS Options Module	 17
3.1 PJSOPT Macro Format and Operands	18
3.2 PJSOPT Module Example	21
 4. PJS Exits	 23
4.1 The PJS Installation Data Format Exit (PJSIDFX)	24
4.1.1 PJSIDFX Input Parameters	24
4.1.2 PJSIDFX Return Codes	24
4.2 The PJS Installation Options Exit (PJSOPTX)	25
4.2.1 PJSOPTX Input Parameters	25
4.2.2 PJSOPTX Return Codes	25

4.3	The PJS Installation Security Exit (PJSSECX)	26
4.3.1	PJSSECX Input Parameters	26
4.3.2	PJSSECX Return Codes	27
4.4	The PJS Installation Submit Exit (PJSSUBX)	28
4.4.1	PJSSUBX Input Parameters.....	28
4.4.2	PJSSUBX Return Codes	29
4.4.3	How to Use the PJSSUBX Exit	29
5.	PJS Security Issues	31
5.1	Securing the PJS Software	31
5.2	Securing the PJS Data Sets	31
5.3	Securing the PJS Request Queue Records	32
5.4	Securing Batch Jobs Submitted by PJS	32
6.	PJS Operator Commands	33
6.1	The START Command	33
6.2	The STOP Command	33
6.3	The SCAN Command	33
7.	PJS Utilities	35
7.1	The PJS Request Queue Initialization (PJSQINIT) Utility	36
7.2	The PJS Request Queue Maintenance (PJSQMNT) Utility	37
A.	PJS TSO Commands	39
B.	PJS Data Areas	41
B.1	The PJS Options Table	41
B.2	The PJS Vector Table	43
B.3	The PJS Request Queue Entry	48
B.4	The PJSQ Queue Control Record	49
B.5	The PJSQ Owner Record	50
B.6	The PJSQ Job Request Record	51
B.7	The PJSQ Calendar Record	55
B.8	The PJSQ Event Record	56
C.	Summary of Changes	57
C.1	Changes for Release 2.1.4	57
C.2	Changes for Release 2.1.3	57
C.3	Changes for Release 2.1	58
	GNU General Public License	61
	GNU Free Documentation License	67

Preface

This manual describes how to install and customize Release 2.1 of the Personal Job Scheduler (PJS).

The person who installs and customizes PJS should have the skills and experience of a Site Administrator or Systems Programmer. Installation requires knowledge of IBM mainframes, the OS/390 environment, JCL, site standards, and site conventions.

During the installation procedure, you can use SMP/E to help install PJS, set prebuilt panels to run under ISPF, or use a security system such as RACF. If you choose any of these options, the manual assumes familiarity with relevant software systems.

If you need more information on how to use this product, please refer to the *Personal Job Scheduler (PJS®) User Guide*. 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.

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

The Personal Job Scheduler (PJS®) enables users at any level of expertise to specify when their jobs are executed; help from operators is not required. For an overview of the PJS system, please refer to Chapters 1 and 2 of the *Personal Job Scheduler (PJS®) User Guide*. This manual assumes familiarity with the concepts and terms introduced in those chapters.

Installation should take between one and three hours. Before you can install PJS, the following software must be installed and operating on your mainframe:

- OS/390® or z/OS®, any supported release.

This manual assumes that you have the skills and experience of a Site Administrator or Systems Programmer. To successfully install PJS, use the information in the following chapters:

- | | |
|-------------------|--|
| Chapter 2 | explains how to install PJS. You can use SMP/E to install PJS, or you can install PJS without SMP/E. Part of the installation procedure will include customizing PJS for use at your site. The installation procedure consists of a set of tasks, one of which will require you to complete a set of subtasks. Each task is covered in a separate section. |
| Chapter 3 | explains the PJSOPT module, which you will create during the installation procedure. |
| Chapter 4 | describes PJS exits, which you can modify as part of the installation procedure. |
| Chapter 5 | describes how to set up your security environment for PJS. |
| Chapter 6 | provides descriptions of PJS operator commands to be used after PJS is installed. |
| Chapter 7 | describes PJS Utilities. |
| Appendix A | lists the PJS TSO commands and the type of access required by each command. |
| Appendix B | contains information on PJS data areas. |

2. How to Install PJS

To install PJS, you must perform the following tasks:

	Task #	Step Description
	1	Obtain the PJS Installation Package
	2	Extract the PJS Installation Package PDS
	3	Review the \$README, \$COPYRT, and \$LICENSE files
	4	Extract the PJS installation files
	5	Extract the PJS documentation files
	6	Allocate the PJS data sets
	7	Install the PJS software (with or without SMP/E)
	8	Add PJS to the Link List or the LPA (Optional)
	9	Authorize the PJS load libraries
	10	Initialize the PJS Request Queue
	11	Allocate the PJS JCL Spool (Optional)
	12	Create the PJS system task procedure
	13	Set up the PJS/TSO interface
	14	Set up the PJS/ISPF Interface
	15	Start the PJS System Task

Sample JCL is provided where appropriate. However, please check all sample jobs before you use them to ensure that site requirements are met.

2.1 Task 1: Obtain the PJS Installation Package

The PJS Installation Package is a single sequential file that contains all the elements of PJS. This file may be obtained as part of a public software distribution tape, such as the MVS CBT Tape, or downloaded from the Internet. If you obtained PJS from the MVS CBT Tape, or another distribution tape, you should refer to the documentation for the distribution tape for specific details of how to extract the PJS Installation Package.

If you download PJS it from the Internet, it may be contained within a Zip file. If this is the case you will need to extract the PJS Installation Package file from the Zip file. In any case whenever the PJS Installation Package, or the Zip file that contains it, is transmitted, typically with FTP, it must be done as BINARY data.

When the PJS Installation Package file (unzipped if necessary) is finally loaded to the OS/390 or z/OS host it is to be installed on, it must be allocated as a physical sequential data set (DSORG=PS), fixed-length records (RECFM=FB), with a logical record length of 80 bytes (LRECL=80), and any appropriate block size. When FTP is used these data set characteristics are typically set using the FTP 'SITE' command. Your installation may also require additional parameters on the 'SITE' command to specify how and where the data set should be allocated. For a complete list of all the options available see the *IBM Communications Server: IP User's Guide* for your release of OS/390 or z/OS.

The following example shows the FTP commands that can be used to send the PJS Installation Package from a PC to an OS/390 or z/OS host:

```
C:\>FTP hostname
Connected to hostname.
220-FTPD1 IBM FTP CS V1R4 at hostname, 16:34:20 on 2004-02-10.
220 Connection will close if idle for more than 20 minutes.
User (hostname:(none)): userid
331 Send password please.
Password: password
230 userid is logged on. Working directory is "userid.".
ftp> CD 'qual'
250 "qual." is the working directory name prefix.
ftp> LCD C:\dir
Local directory now C:\dir.
ftp> QUOTE SITE RECFM=FB LRECL=80 BLKSIZE=0 TRACKS PRI=300 SEC=75
200 SITE command was accepted
ftp> BINARY
200 Representation type is Image
ftp> PUT PJS214.XMIT
200 Port request OK.
125 Storing data set qual.PJS214.XMIT
250 Transfer completed successfully.
ftp: 10371120 bytes sent in 9.26Seconds 1119.51Kbytes/sec.
ftp> QUIT
221 Quit command received. Goodbye.
```

2.2 Task 2: Extract the PJS Installation Package PDS

The PJS Installation Package contains a partitioned data set that has been formatted as a TSO TRANSMIT file. This PDS, called the PJS Installation Package PDS may be extracted with the TSO 'RECEIVE' command with the 'INDATASET' parameter.

The following example shows the TSO 'RECEIVE' command that can be used to extract the PJS Installation Package PDS:

```
READY
RECEIVE INDATASET('qual.PJS214.XMIT')
  INMR901I Dataset qual.DDNAME.PACKAGE from TCH03 on SYSTEME
  INMR906A Enter restore parameters or 'DELETE' or 'END' +
DATASET('qual.PJS214.PACKAGE') NEW
  INMR001I Restore successful to dataset 'qual.PJS214.PACKAGE'
READY
```

Your installation may also require additional parameters on the 'RECEIVE' command prompt response to specify how and where the data set should be allocated. For a complete list of all the options available see the *TSO/E Command Reference* for your release of OS/390 or z/OS.

The following table lists the members of the PJS Installation Package PDS:

Member Name	Description	Format
\$COPYRT	PJS Copyright Notice	Text
\$LICENSE	GNU General Public License	Text
\$LICPUB	GNU Free Documentation License	Text
\$README	“Getting Started” Instructions	Text
EXTRACT	Sample job to extract the PJS installation files	Text
PJSDOC	PJS Documentation	Zip
INSTALL	Installation JCL Library	PDS
SMPMCS	SMP/E MCS Statements	Seq
JCLIN	SMP/E JCLIN	PDS
MACLIB	PJS Macro Library	PDS
SRCLIB	PJS Source Library	PDS
MODLIB	PJS Modules Library	PDS
TSOHELP	PJS TSO Help Library	PDS
ISPFPNL	PJS ISPF Panels Library	PDS
ISPFMSG	PJS ISPF Messages Library	PDS
ISPFTBL	PJS ISPF Tables Library	PDS
SAMPLIB	PJS Samples Library	PDS
UCRLIB	PJS User Contributed Routines	PDS

Text = Text file

Zip = Zip file in TSO XMIT format

Seq = Sequential Data Set in TSO XMIT format

PDS = Partitioned Data Set in TSO XMIT format

The \$COPYRT, \$LICENSE, \$LICPUB, and \$README members are text files that contain important information about PJS that should be reviewed before completing the installation. The EXTRACT member contains a sample job that can be used to create the PJS installation files from the PJS Installation Package. Each of the other members contains one of the PJS installation files that has been formatted as a TSO TRANSMIT file.

2.3 Task 3: Review the \$COPYRT, \$LICENSE, and \$README Files

After extracting the PJS Installation Package you should carefully read the \$COPYRT and \$LICENSE members. These contain the copyright notice and license terms for using PJS. If you do not agree to these terms you will not be authorized to use PJS.

PJS is licensed under the GNU General Public License version 2, or (at your option) any later version, as published by the Free Software Foundation. This license is widely used for many open-source software products, including Linux. A copy of the GNU GPL is also included in an appendix of this book. For more information about the GNU GPL see the Free Software Foundation website at <http://www.gnu.org/licenses/>.

You should also read the \$README member. This contains the latest information that might not have been included in the documentation.

2.4 Task 4: Extract the PJS Installation Files

The EXTRACT member of the PJS Installation Package PDS contains a sample job to extract the PJS installation data sets. This job will run TSO as a background job and use the TSO 'RECEIVE' command with the 'INDATASET' parameter to extract each of the PJS Installation Files.

This job should be tailored to suit your installation standards, paying particular attention to the job card, and the allocation parameters on the TSO 'RECEIVE' command prompt responses. For a complete list of all the options available see the *TSO/E Command Reference* for your release of OS/390 or z/OS.

The extract process will create the following PJS Installation Files:

Data Set Name	Description	Format
qual.PJS214.PJSDOC	PJS Documentation	Zip
qual.PJS214.INSTALL	Installation JCL Library	PDS
qual.PJS214.SMPMCS	SMP/E MCS Statements	Seq
qual.PJS214.NPJ2104.F1	SMP/E JCLIN	PDS
qual.PJS214.NPJ2104.F2	PJS Macro Library	PDS
qual.PJS214.NPJ2104.F3	PJS Source Library	PDS
qual.PJS214.NPJ2104.F4	PJS Modules Library	PDS
qual.PJS214.NPJ2104.F5	PJS TSO Help Library	PDS
qual.PJS214.NPJ2104.F6	PJS ISPF Panels Library	PDS
qual.PJS214.NPJ2104.F7	PJS ISPF Messages Library	PDS
qual.PJS214.NPJ2104.F8	PJS ISPF Tables Library	PDS
qual.PJS214.NPJ2104.F9	PJS Samples Library	PDS
qual.PJS214.UCRLIB	PJS User Contributed Routines	PDS

2.5 Task 5: Extract the PJS Documentation Files

If you download PJS it from the Internet as a Zip file, the PJS documentation may have been included as a separate file within that Zip file. If this is not the case you will need to get the PJS documentation from the PJS Installation Package.

In Task 4 you should have created a file named '*qual.PJS214.PJSDOC*'. This is a variable-length record sequential data set that is a Zip file containing the PJS documentation files. To extract and view these files you will need to download them to a workstation, typically with FTP. This transfer must be done in BINARY mode.

The following example shows the FTP commands that can be used to receive the PJS documentation Zip file to a PC from an OS/390 or z/OS host:

```
C:\>FTP hostname
Connected to hostname.
220-FTPD1 IBM FTP CS V1R4 at hostname, 16:34:20 on 2004-02-10.
220 Connection will close if idle for more than 20 minutes.
User (hostname:(none)): userid
331 Send password please.
Password: password
230 userid is logged on. Working directory is "userid".
ftp> CD 'qual.PJS214'
250 "qual.PJS214." is the working directory name prefix.
ftp> LCD C:\dir
Local directory now C:\dir.
ftp> BINARY
200 Representation type is Image
ftp> GET PJSDOC PJSDOC.ZIP
200 Port request OK.
125 Storing data set qual.PJS214.PJSDOC
250 Transfer completed successfully.
ftp: 783588 bytes received in 1.12Seconds 699.01Kbytes/sec.
ftp> QUIT
221 Quit command received. Goodbye.
```


Once the file is on your workstation you will need to extract the documentation files using WinZip, or any other program that extracts Zip files. Once the documents have been extracted the following documents will be available:

File Name	Description	Format
PJSINST.PDF	PJS Installation Guide	PDF
PJSMSG.PDF	PJS Messages and Codes	PDF
PJSUSER.PDF	PJS User's Guide	PDF

PDF = Adobe Acrobat Portable Document File. To view these files you will need the free Adobe Acrobat Reader, available at <http://www.adobe.com/products/acrobat/readstep2.html>.

2.6 Task 6: Allocate the PJS Data Sets

Minimum allocation requirements for the PJS distribution and target libraries are listed in the following tables. To allow for maintenance and expansion, add extra space to each data set.

The member ALLOC in the INSTALL library contains a sample job to allocate these data sets. This job should be tailored to suit your installation standards, paying particular attention to the job card, volume serial numbers, and unit names.

PJS Distribution Libraries

File Name	Data Set Name	File Description	RECFM	LRECL	# of 3390 Tracks	# of Dir Blks
APJSMAC	<i>qual.PJS.APJSMAC</i>	PJS Macro DLIB	FB	80	39	2
APJSSRC	<i>qual.PJS.APJSSRC</i>	PJS Source DLIB	FB	80	*	*
APJSMOD	<i>qual.PJS.APJSMOD</i>	PJS Module DLIB	U	0	107	16
APJSHELP	<i>qual.PJS.APJSHELP</i>	PJS TSO Help DLIB	FB	80	38	5
APJSPLIB	<i>qual.PJS.APJSPLIB</i>	PJS ISPF Panels DLIB	FB	80	256	22
APJSMLIB	<i>qual.PJS.APJSMLIB</i>	PJS ISPF Messages DLIB	FB	80	36	7
APJSTLIB	<i>qual.PJS.APJSTLIB</i>	PJS ISPF Tables DLIB	FB	80	2	1
APJSSAMP	<i>qual.PJS.APJSSAMP</i>	PJS Samples DLIB	FB	80	114	2

If you do not plan to use SMP/E to install PJS, just allocate the APJSMOD data set. Because this library is required only during the installation process, you can delete APJSMOD after you install PJS.

If you use SMP/E to install PJS, allocate all of these data sets. Do not delete APJSMOD after you complete the PJS installation.

PJS Target Libraries

File Name	Data Set Name	File Description	RECFM	LRECL	# of 3390 Tracks	# of Dir Blks
PJSMAC	<i>qual.PJS.MACLIB</i>	PJS Macro Library	FB	80	39	2
PJSSRC	<i>qual.PJS.SRCLIB</i>	PJS Source Library	FB	80	*	*
PJSCMD	<i>qual.PJS.COMDLIB</i>	PJS TSO Command Library	U	0	43	11
PJSLINK	<i>qual.PJS.LINKLIB</i>	PJS Link Library (Common Routines)	U	0	21	3
PJSLOAD	<i>qual.PJS.LOADLIB</i>	PJS Load Library (System Task and Utilities)	U	0	9	2
PJSHELP	<i>qual.PJS.TSOHELP</i>	PJS TSO Help Library	FB	80	38	5
PJSPLIB	<i>qual.PJS.ISPFPNL</i>	PJS ISPF Panels Library	FB	80	256	22
PJSMLIB	<i>qual.PJS.ISPFMSG</i>	PJS ISPF Messages Library	FB	80	36	7
PJSTLIB	<i>qual.PJS.ISPFTBL</i>	PJS ISPF Tables Library	FB	80	2	1
PJSSAMP	<i>qual.PJS.SAMPLIB</i>	PJS Samples Library	FB	80	114	2

You can combine the PJSCMD, PJSLINK, and PJSLOAD data sets, or combine these data sets with other data sets by changing the appropriate JCL DD statements and SMP/E DDDEFs. For example, the PJSCMD and PJSLINK DD names can reference SYS1.LPALIB, and the PJSLOAD DD name can reference a common APF authorized non-link-list data set.

The load modules contained in the PJSCMD and PJSLINK libraries are reentrant, so they can be included in the system Link Pack Area (LPA). If you decide to include these reentrant load modules in the LPA, you must perform an IPL with CLPA after the installation to load the modules into the LPA. These modules use about 12K of LPA (below the 16M storage line), and about 192K of ELPA (above the 16M storage line).

2.7 Task 7: Install the PJS Software

If you are using SMP/E to install PJS, complete the set of subtasks in Subsection 2.7.1. If you are not using SMP/E to install PJS, complete the set of subtasks in Subsection 2.7.2.

2.7.1 Installation with SMP/E

You can use the SMP/E ISPF Dialogs or the sample jobs to help finish the following subtasks:

2.7.1.1 SMP/E Subtask 1: Prepare the SMP/E Zones

PJS can be installed into an existing SMP/E zone, if there are no name conflicts. It can also be installed into a separate zone. If you wish to install PJS into an existing zone, the member SMPDDDEF in the INSTALL library provides a sample job to define the DDDEFs required by PJS. If you wish to install PJS into a new zone, the member DEFCSI in the INSTALL library provides a sample job to create and define a new SMP/E CSI for PJS.

2.7.1.2 SMP/E Subtask 2: Receive, Apply, Accept PJS Function SYSMOD

Use the following procedure to receive, apply, and accept the PJS function SYSMOD:

1. Receive the PJS function SYSMOD (FMID NPJ2104) into the SMP/E Global Zone. The member SMPRECF in the INSTALL library provides a sample job to receive the PJS function SYSMOD. This job should be tailored to suit your installation standards, paying particular attention to the job card, and the SMP/E CSI data set name.
2. Apply the PJS function SYSMOD (FMID NPJ2104) to the SMP/E Target Zone. The member SMPAPLYF in the INSTALL library provides a sample job to apply the PJS function SYSMOD. This job should be tailored to suit your installation standards, paying particular attention to the job card, the SMP/E CSI data set name, and the Target Zone name.
3. Accept the PJS function SYSMOD (FMID NPJ2104) into the SMP/E Distribution Zone. The member SMPACPTF in the INSTALL library provides a sample job to accept the PJS function SYSMOD. This job should be tailored to suit your installation standards, paying particular attention to the job card, the SMP/E CSI data set name, and the Distribution Zone name.

2.7.1.3 SMP/E Subtask 3: Receive and Apply User Exit USERMODs (Optional)

Note: User exits provide the means to tailor PJS to better fit site requirements. Even if you do not plan to change user exits and skip this subtask it is a good idea to read the information contained in Chapter 4.

Chapter 4 contains information on the PJS Installation Data Format Exit, the PJS Options Exit, the PJS Security Exit, and the PJS Installation Submit Exit. If you plan to make use of any of these user exits, you can use the sample exits provided or use the information in Chapter 4 to write your own exit routines.

If PJS is being upgraded from an earlier release, the exit source should be reviewed for any required changes, then reinstalled. Do not attempt to use an existing object or load module.

The module source should be placed into an appropriate USERMOD and received into the SMP/E Global Zone. Apply the USERMODs to the SMP/E Target Zone.

The member USERMOD in the INSTALL library provides a sample SMP/E USERMOD for modifying a PJS module. This member should be used as a base, by copying it to another data set and/or member, editing the SYSMOD ID, and adding the source code for the module being modified.

The member SMPRECU in the INSTALL library provides a sample job to receive a USERMOD for PJS. This job should be tailored to suit your installation standards, paying particular attention to the job card, and the SMP/E CSI data set name.

The member SMPAPLYU in the INSTALL library provides a sample job to apply a USERMOD for PJS. This job should be tailored to suit your installation standards, paying particular attention to the job card, the SMP/E CSI data set name, and the USERMOD ID.

Note: We recommend that you do *not* accept the USERMODs for user exits into the SMP/E Distribution Zone. If you need more information, please refer to the *SMP/E User's Guide*.

2.7.1.4 SMP/E Subtask 4: Receive and Apply the PJSOPT USERMOD

Create a USERMOD for the PJS Options Module, which is contained in the PJSOPT module. Information on how to create source code for this module is contained in Chapter 3.

If PJS is being upgraded from an earlier release, the PJSOPT source should be reviewed for any required changes, then reinstalled. Do not attempt to use an existing object or load module.

The module source should be placed into an appropriate USERMOD and received into the SMP/E Global Zone. Apply the PJSOPT USERMOD to the SMP/E Target Zone.

The member USERMOD in the INSTALL library provides a sample SMP/E USERMOD for modifying a PJS module. This member should be used as a base, by copying it to another data set and/or member, editing the SYSMOD ID, and adding the source code for the module being modified.

The member SMPRECU in the INSTALL library provides a sample job to receive a USERMOD for PJS. This job should be tailored to suit your installation standards, paying particular attention to the job card, and the SMP/E CSI data set name.

The member SMPAPLYU in the INSTALL library provides a sample job to apply a USERMOD for PJS. This job should be tailored to suit your installation standards, paying particular attention to the job card, the SMP/E CSI data set name, and the USERMOD ID.

Note: We recommend that you do *not* accept the USERMOD for PJSOPT into the SMP/E Distribution Zone. If you need more information, please refer to the *SMP/E User's Guide*.

After you finish this subtask, please refer to Section 2.8 to continue installing PJS.

2.7.2 Installation without SMP/E

Perform the following subtasks:

2.7.2.1 Non-SMP/E Subtask 1: Copy PJS from the PJS Installation Files

The member COPYPJS in the INSTALL library provides a sample job to copy PJS from the PJS Installation Files to the PJS target data sets using IEBCOPY.

2.7.2.2 Non-SMP/E Subtask 2: Link-Edit the PJS Software

The TSO command procedures are link-edited into the PJSCMD library, the common reentrant subroutines are link-edited into the PJSLINK library, and the non-reentrant utilities and the PJS System Task programs are link-edited into the PJSLOAD library. The member LINKPJS in the INSTALL library provides a sample job to link-edit the PJS software.

2.7.2.3 Non-SMP/E Subtask 3: Assemble and Link-Edit User Exits (Optional)

Note: User exits provide the means to tailor PJS to better fit site requirements. Even if you do not plan to change user exits and skip this subtask it is a good idea to read the information contained in Chapter 4.

Chapter 4 contains information on the PJS Installation Data Format Exit, the PJS Options Exit, the PJS Security Exit, and the PJS Installation Submit Exit. If you plan to make use of any of these user exits, you can use the sample exits provided or use the information in Chapter 4 to write your own exit routines.

If PJS is being upgraded from an earlier release, the exit source should be reviewed for any required changes, then reassembled and link-edited. Do not attempt to use an existing object or load module.

The member ASMEXIT in the INSTALL library provides a sample job to assemble and link-edit an exit routine for PJS. This job should be tailored to suit your installation standards, paying particular attention to the job card, and the data set names.

2.7.2.4 Non-SMP/E Subtask 4: Assemble and Link-Edit PJSOPT

Create a PJS Options Module (PJSOPT). Details on how to create the source code for the PJSOPT module are contained in Chapter 3.

If PJS is being upgraded from an earlier release, the PJSOPT source should be reviewed for any required changes, then reassembled and link-edited. Do not attempt to use an existing object or load module.

The member ASMOPT in the INSTALL library provides a sample job to assemble and link-edit the PJS Options Module. This job should be tailored to suit your installation standards, paying particular attention to the job card, and the data set names.

2.8 Task 8: Add PJS to the Link List or the LPA (Optional)

It's a good idea to place the PJSCMD library and the PJSLINK library either in the system link list or the link pack area. You can place the PJSLOAD library in the system link list, but there's no good reason to do so; you cannot place PJSLOAD in the LPA.

For more information and instructions, refer to the *MVS Initialization and Tuning Reference*. An IPL (with CLPA) may be required to make this change effective.

2.9 Task 9: Authorize the PJS Load Libraries

The PJS System Task should run APF authorized. The main routines used by the PJS System Task are in the PJSLOAD library; it also uses several routines from the PJSLINK (common routines) library. These libraries should be APF authorized. If these libraries are placed on the system link list, and the system link list is authorized, they may acquire authorization from that fact. If the PJSLINK library is placed in the LPA, those routines will be considered authorized. Otherwise, the data sets in which these libraries are placed, must be listed in the IEAAPFnn member of SYS1.PARMLIB. If PJS runs without authorization, PJS cannot send messages to its users. In addition, the PJS Security Exit must be authorized if you plan to use it.

Refer to the *MVS Initialization and Tuning Reference* for detailed instructions. An IPL may be required to make this change effective.

2.10 Task 10: Initialize the PJS Request Queue

Note: If you are upgrading PJS, you can omit this step. If this is a new installation, you must complete this task.

The PJS Request Queue contains records defining the Job Requests, Calendars, and Events specified by PJS users, as well as other control records.

The PJS Request Queue is a VSAM KSDS. The PJS Request Queue must be defined with RECSZ(144,4084), KEYS(17,0), and SHR(2). Most other data set attributes may be changed to optimize VSAM processing. For most installations, 1 or 2 cylinders of space should be sufficient.

The PJS Request Queue is defined with IDCAMS and initialized with the PJSQINIT program. PJSQINIT is a utility program that initializes the PJS Request Queue. It has no input parameters. A DD statement with a DDNAME of PJSQ is required, and should point to the PJS Request Queue data set. A DD statement with a DDNAME of SYSOUT is also required, and should point to the message data set; in most cases you can specify **SYSOUT=***.

The member DEFPJSQ in the INSTALL library provides a sample job to define and initialize the PJS Request Queue data set. This job should be tailored to suit your installation standards, paying particular attention to the job card, the PJS Request Queue data set name, the volume serial number, and the allocation quantity.

For more information on PJSQINIT, please refer to Chapter 7.

2.11 Task 11: Allocate the PJS JCL Spool (Optional)

Note: If this is a maintenance installation (the same release), you can omit this task.

If the PJS JCL Save feature is to be used, the PJS JCL Spool data set must be allocated. The JCL to be saved will be copied to the PJS JCL Spool by the PJS TSO commands and the PJS ISPF interface.

The PJS JCL Spool is a partitioned data set. The PJS JCL Spool must be defined with RECFM=FB, LRECL=80, and any suitable blocksize. Each JCL member saved has a PDS directory entry. Since each directory entry has 22 bytes of user data, 7 directory entries will fit in each directory block.

The member DEFSPPOOL in the INSTALL library provides a sample job to allocate the PJS Spool data set. This job should be tailored to suit your installation standards, paying particular attention to the job card, the PJS JCL Spool data set name, the unit name, the volume serial number, and the allocation quantity.

2.12 Task 12: Create the PJS System Task Procedure

Note: If PJS is already installed, you can omit this task. If this is a new installation, you must complete this task.

Place the start-up procedure for the PJS System Task into a system procedure library. A sample procedure is in member PJSPROC in the INSTALL library. This procedure should be tailored to suit your installation standards.

The PJSLOAD library should be included in the STEPLIB concatenation. (Although it is possible to include the PJSLOAD library in the system link list, this is not generally recommended.) The PJSLINK library must be concatenated to the STEPLIB data set, included in the system link list, or placed in the system link pack area.

A DD statement with a DDNAME of SYSOUT is required, and should point to the message data set; in most cases you can specify **SYSOUT=***.

2.13 Task 13: Set Up the PJS/TSO Interface

The PJS/TSO interface must be made available to the user. The PJSCMD library and PJSLINK library must be concatenated to the STEPLIB data set, included in the system link list, or placed in the system link pack area. The PJSHELP library should be concatenated to the SYSHELP data set.

If you have a security system that restricts TSO commands (like ACF2), you may need to define the PJS TSO commands to your security system. The PJS TSO commands that must be defined are listed in Appendix A.

For more information, please refer to *TSO/E Customization* and your security system documentation.

2.14 Task 14: Set Up the PJS/ISPF Interface (Optional)

Note: If you do not plan to use the ISPF interface, skip this task.

The PJS ISPF panels, messages, and tables must be made available to ISPF, and a means of selecting PJS must be provided to the user. Each of these may be accomplished in several ways. The best way depends on how your installation manages ISPF. Detailed and authoritative information on each method can be found in the *ISPF Dialog Management Guide and Reference*. The particular documents required may depend on your release of ISPF.

There are several ways to make the PJS ISPF panels, messages, and tables available to ISPF:

- Install the PJS panels, messages and tables in existing ISPF libraries (such as SYS1.ISPPLIB, SYS1.ISPMLIB, and SYS1.ISPTLIB).
- Concatenate the PJS libraries to the data sets already allocated to the ISPF libraries (ISPPLIB, ISPMLIB, and ISPTLIB). This could be done either in the TSO procedure, or by a CLIST or REXX exec that executes prior to ISPF.
- Allocate the PJS libraries using the ISPF LIBDEF service. This may require a CLIST or REXX exec to execute prior to the ISPF SELECT for PJS.

Generally, to call the PJS ISPF interface, the following ISPF SELECT is required:

```
SELECT PGM(PJSISPF) PARM(option) NEWAPPL(PJS) NOCHECK
```

where *option* is the initial PJS option to be selected.

To modify an ISPF menu to call the PJS/ISPF interface, the following statements should be included in the)PROC section of the menu panel definition:

```
&ZQ = TRUNC(&ZCMD,'.')
&ZTRAIL = .TRAIL
&ZSEL = TRANS( &ZQ
               .
               .
               .
               n, 'PGM(PJSISPF) PARM(&ZTRAIL) NEWAPPL(PJS) NOCHECK'
               .
               .
               .
               *, '?' )
```

2.15 Task 15: Start the PJS System Task

Once the PJS system is installed and customized, the PJS System Task can be started. Procedures for starting, stopping, and controlling the PJS System Task are in Chapter 6.

It is recommended that the start command for the PJS System Task be placed in the MVS automatic start-up commands list (member COMMNDxx of SYS1.PARMLIB). For detailed information on how to do this, see the *MVS Initialization and Tuning Reference*.

3. The PJS Options Module

One of the PJS installation subtasks requires you to create the PJS Options Module (PJSOPT). PJSOPT is a non-executable CSECT that defines certain constants used by the PJS system. Use the following format to build the PJSOPT module:

```
PJSOPT macro specification  
END
```

A PJSOPT module is created by the PJSOPT macro. An assembler END statement must be included at the end.

The PJSOPT module should be link-edited into a load module of the same name and placed into the PJSLINK library. Specify the RENT attribute; it will be assigned the attributes AMODE=31, and RMODE=ANY.

3.1 PJSOPT Macro Format and Operands

The PJSOPT macro has the following format:

```

PJSOPT  PJSQDSN=data-set-name
        [ ,CMPTIME={ 7 | nnn } ]
        [ ,DATEFMT={ MDY | DMY } ]
        [ ,DISTIME={ 30 | nnn } ]
        [ ,ENQNAME={ PJSQUEUE | qname } ]
        [ ,ERRTIME={ 7 | nnn } ]
        [ ,ETBSZ={ 10000 | nnnnn } ]
        [ ,ISPFQSZ={ 1000 | nnnnn } ]
        [ ,JCLSAVE={ NO | OPT | DFLT | REQ } ]
        [ ,JCLSDSN=data-set-name ]
        [ ,JCLSLIM=( { 500 | nnnnn } , { 10000 | nnnnn } ) ]
        [ ,OTBSZ={ 1000 | nnnnn } ]
        [ ,PTBSZ={ 100 | nnnnn } ]
        [ ,RDRCLS={ A | class } ]
        [ ,RETRY=( { 12 | nnnnn } , ( { 5 | nnnnn } ) ) ]
        [ ,RRTBSZ={ 1000 | nnnnn } ]
        [ ,SCANINT={ 15 | nnn } ]
        [ ,STBSZ={ 100000 | nnnnn } ]
        [ ,TMPBLK={ 8880 | nnnnn } ]
        [ ,TMPPRIM={ 10 | nnnnn } ]
        [ ,TMPSEC={ 10 | nnnnn } ]
        [ ,TMPUNIT=unit ]
        [ ,TSOAUTH={ YES | NO } ]

```

Parameter descriptions follow in displayed order.

PJSQDSN=*data-set-name* is the data set name of the PJS Request Queue. This parameter is required; there is no default.

CMPTIME=*nnn* specifies the number of days a job is held in COMPLETED status before it is deleted. A value of **0** indicates no limit. The default is **7** days.

DATEFMT={MDY|DMY}

specifies the format in which PJS is to display dates, and accept dates for input. **MDY** means dates are displayed and input in the *mm/dd/yyyy* format (commonly used in the U.S.). **DMY** means dates are displayed and input in the *dd/mm/yyyy* format (commonly used in Europe). The default is **MDY**.

DISTIME=nnn

specifies the number of days a job is held in **DISABLED** status before it is deleted. A value of **0** indicates no limit. The default is **30** days.

ENQNAME=qname

specifies the 8-character **QNAME** that PJS will use when it serializes resources. The default is **PJSQUEUE**.

ERRTIME=nnn

specifies the number of days a job is held in **ERROR** status before it is deleted. A value of **0** indicates no limit. The default is **7** days.

ETBLSZ=nnnnn

specifies the number of entries to allocate for the PJS Request Queue Maintenance Utility Event Table. One entry is required for each Event in the PJS Request Queue. Each entry requires 17 bytes of virtual storage. The default is **10000** entries.

ISPFQSZ=nnnnn

specifies the number of entries to allocate for the PJS ISPF Interface List Job Requests Table, List Calendars Table, or List Events Table. One entry is required for each Job Request, Calendar, or Event selected by a list dialog. If this value is exceeded, the list is truncated and a system message is displayed. Each entry requires up to 256 bytes. The default is **1000** entries.

JCLSAVE={NO|OPT|DFLT|REQ}

specifies whether the PJS JCL Save Feature will be enabled. **NO** means that the PJS JCL Save Feature cannot be used. **OPT** means that the PJS JCL Save Feature can be used, but the default is **NOSAVE**. **DFLT** means that the PJS JCL Save Feature can be used, and the default is **SAVE**. **REQ** means that the PJS JCL Save Feature must be used. The default is **NO**.

JCLSDSN=data-set-name

is the data set name of the PJS JCL Spool data set. This parameter is required, unless **JCLSAVE=NO** is specified or defaulted. There is no default.

JCLSLIM=(nnnnn,nnnnn)

specifies the maximum number of JCL records that can be stored in the PJS JCL Spool. The first value is the maximum for a single Job Request. The second value is the maximum for all Job Requests for an Owner. The default is **500** records per Job Request, and **10000** records per Owner.

OTBLSZ=nnnnn

specifies the number of entries to allocate for the PJS Request Queue Maintenance Utility Owner Table. One entry is required for each Owner in the PJS Request Queue. Each entry requires 8 bytes of virtual storage. The default is **1000** entries.

PTBLSZ=nnnnn

specifies the number of entries to allocate for the PJS System Task Post/Reset Event Table. One entry is required for each Event posted within a PJS Short Request Queue Scan Interval (usually about 1 minute). Each entry requires 24 bytes of virtual storage. The default is **100** entries.

RDRCLS= <i>class</i>	specifies the SYSOUT Class to be used when allocating an internal reader. The default is A .
RETRY= (<i>nnnnnn,nnnnn</i>)	specifies how PJS os to retry a job submission when a temporary error (i.e. JCL data set in use) occurs. The first value is the maximum number of times the submit is to be retried before placing the request in the ERROR status. The second value is the time interval (in minutes) between each retry attempt. RETRY=(0,0) means that no retry is to be attempted. The default is RETRY=(12,5) .
RRTBLSZ= <i>nnnnnn</i>	specifies the number of entries to allocate for the PJS System Task Ready Request Table. One entry is required for each Job Request scheduled within a PJS Long Request Queue Scan Interval, a value set by the SCANINT= keyword discussed below. Each entry requires 32 bytes of virtual storage. The default is 1000 entries.
SCANINT= <i>nnnnnn</i>	specifies the number of minutes in the PJS Long Request Queue Scan Interval. This interval is the time between Request Queue scans by the PJS System Task. Shorter times minimize the size of the PJS System Task Ready Request Table. Longer times reduce system overhead. In PJS Release 1.0, a small value meant a shorter time delay for new or changed requests with run times only slightly after the add or change time. In PJS Release 2.0, this is no longer an advantage. The default is 15 minutes.
STBLSZ= <i>nnnnnn</i>	specifies the number of entries to allocate for the PJS Request Queue Maintenance Utility Spool Directory Table. One entry is required for each Job Request for which the JCL is saved in the PJS JCL Spool. Each entry requires 80 bytes of virtual storage. The default is 1000 entries.
TMPBLK= <i>nnnnnn</i>	specifies the block size to be used when allocating a temporary data set to be used for listing or browsing JCL from the PJS JCL Spool. The default is 8880 .
TMPPRIM= <i>nnnnnn</i>	specifies the number of blocks for the primary space allocation for a temporary data set to be used for listing or browsing JCL from the PJS JCL Spool. The default is 10 blocks.
TMPSEC= <i>nnnnnn</i>	specifies the number of blocks for the secondary space allocation for a temporary data set to be used for listing or browsing JCL from the PJS JCL Spool. The default is 10 blocks.
TMPUNIT= <i>unit</i>	specifies the unit name to be used when allocating a temporary data set to be used for listing or browsing JCL from the PJS JCL Spool. A null value indicates that the user's default TSO allocation unit is to be used. The default is null.

TSOAUTH={YES|NO} specifies whether the PJS ISPF interface is to invoke the PJS ISPF Interface TSO commands from an authorized environment. You may have to specify **YES** to secure the PJS Request Queue with RACF (or some other security interface) Program Access to Data Set; you may also need to specify the commands in the TSO Authorized Command List. (The PJS TSO Commands are listed in Appendix A). **NO** means that the PJS ISPF Interface Routine will be invoked from an unauthorized environment; this is a more efficient call. The default is **YES**.

3.2 PJSOPT Module Example

The following example shows how to use the PJSOPT and PJSAUTH macros to create the PJS Options Module (PJSOPT):

```

.....+.....1.....+.....2.....+.....3.....+.....4.....+.....5.....+.....6.....+.....7.....+
      PJSOPT  PJSQDSN=PJS.QUEUE,                                *
              EVENTSZ=256,                                       *
              SCANINT=60
      END

```

The name of the PJS Request Queue is specified as PJS.QUEUE, 256 event table entries are defined, and the scan interval time is set to 60 minutes. Two CPUs are authorized.

4. PJS Exits

One of the optional installation subtasks in Chapter 2 enables you to modify PJS user exits so that PJS meets site requirements. The following exits will be discussed:

Module Name	Module Description (Relevant Section)
PJSIDFX	PJS Installation Data Format Exit (Section 4.1)
PJSOPTX	PJS Options Exit (Section 4.2)
PJSSECX	PJS Security Exit (Section 4.3)
PJSSUBX	PJS Submit Exit (Section 4.4)

The PJS Installation Data Format Exit enables you to format the Installation Data Area from a PJS Job Request Record for display.

The PJS Options Exit enables you to dynamically set options at installation time.

The PJS Security Exit enables you to define and establish access control to PJS request records.

The PJS Submit Exit enables you to automatically modify JCL submitted through PJS and, if needed, to suppress the job.

The default exits perform no significant actions and have no adverse side effects. Even if you do not intend to modify any of these exits, we recommend that you read about them. Please pay particular attention to the section on the PJS Security Exit (PJSSECX); this section and Chapter 5 both contain a great deal of information on PJS site security issues.

Several sample exits are provided in the PJSSAMP library. Each exit is documented in the comments at the start of the program source.

4.1 The PJS Installation Data Format Exit (PJSIDFX)

PJSIDFX is an exit routine that enables the installation to format the data in the installation data area of a PJS Job Request Record for display. For an example of how this information can be displayed, please refer to Chapter 3 in the *Personal Job Scheduler (PJS) User Guide*. This exit will be called by the PJS TSO command processors and the PJS ISPF interface when a job request record is formatted for display.

The PJSIDFX module must be link-edited with the RENT, AMODE=31, and RMODE=ANY attributes specified. The load module should be placed into the PJSLINK library.

The PJSIDFX exit is called using standard IBM linkage conventions. Register 15 will point to the program entry point address. The exit should return to the address passed in register 14. Register 13 will point to a 72-byte save area into which the exit should save the caller's registers. On return, the caller's registers (except register 15, which is set to the return code) must be restored.

4.1.1 PJSIDFX Input Parameters

Register 1 will point to the following parameter list:

- Parm 1 - Address of the PJSVT (PJS Vector Table)
- Parm 2 - Address of the PJS Job Request Record being formatted
- Parm 3 - Address of the area in which the exit is to place the formatted data

If the Input PJS Job Request Record address (Parm 2) is zero, the exit should place a heading for the installation data into the Formatted Installation Data Area (Parm 3). If the input PJS Job Request Record address (Parm 2) is non-zero, the exit should place the formatted installation data for the record into the Formatted Installation Data Area (Parm 3). The PJS Job Request Record must **not** be changed by the exit. For more information on the PJS Vector Table and the Job Request Record, please refer to Appendix B.

The format of the Formatted Installation Data Area is:

- Bytes 0 - 1 Length of text area (set by calling program)
- Bytes 2 - 3 Length of formatted text (to be set by the exit)
- Bytes 4 - n Text area

The first length is the length of the text area (not including the length fields) and is set by the calling program. The second length is the actual length of the formatted text, and is to be set by this routine. The formatted text length should generally be set to the same length for the heading and all data lines, since this length is used to determine break points for multi-line data and scroll lengths for the ISPF List Job Request dialog.

4.1.2 PJSIDFX Return Codes

On return, register 15 should be set to a return code of zero.

4.2 The PJS Installation Options Exit (PJSOPTX)

PJSOPTX is an exit routine that enables the installation to dynamically set installations options at execution time. For example, this exit can enable you to run multiple copies of PJS in some specialized environments.

This exit will be called by the PJS routines when the PJS Installation Options Module is loaded. The information in the options module is copied into the PJS Vector Table (PJSVT) and the exit called. The exit can then test any system control blocks available to it to change the installation options as desired.

The PJSOPTX module must be link-edited with the RENT, AMODE=31, and RMODE=ANY attributes specified. The load module should be placed into the PJSLINK library.

The PJSOPTX exit is called using standard IBM linkage conventions. Register 15 will point to the program entry point address. The exit should return to the address passed in register 14. Register 13 will point to a 72-byte save area into which the exit should save the caller's registers. On return, the caller's registers (except register 15, which is set to the return code) must be restored.

4.2.1 PJSOPTX Input Parameters

Register 1 will point to the following parameter list:

Parm 1 - Address of the PJSVT (PJS Vector Table)

For more information on the PJS Vector Table, please refer to Appendix B.

4.2.2 PJSOPTX Return Codes

On return, register 15 should be set to a return code of zero.

4.3 The PJS Installation Security Exit (PJSSECX)

Note: For more information on PJS security issues, please refer to Chapter 5.

PJSSECX is an exit routine that enables the installation to secure the PJS Request Queue and provide access control to the PJS functions and the request records. This exit will be called by several of the PJS programs at strategic points that can enable the installation to provide complete security. The specific calling point is identified by the function code parameter.

It is recommended that some type of security exit be written. The default exit will always approve all requests - it provides no security.

The PJSSECX module must be link-edited with the RENT, AMODE=31, and RMODE=ANY attributes specified. The load module should be placed into the PJSLINK library.

The PJSSECX exit is called using standard IBM linkage conventions. Register 15 will point to the program entry point address. The exit should return to the address passed in register 14. Register 13 will point to a 72-byte save area into which the exit should save the caller's registers. On return, the caller's registers (except register 15, which is set to the return code) must be restored.

4.3.1 PJSSECX Input Parameters

Register 1 will point to the following parameter list:

Parm 1 - Pointer to a 1-byte function code. This will contain one of the following values:

X'01' - Begin TSO Command Processing	X'82' - Add Job Request Record
X'02' - End TSO Command Processing	X'83' - Update Job Request Record
X'11' - Begin ISPF Interface Processing	X'84' - Delete Job Request Record
X'12' - End ISPF Interface Processing	X'91' - Read Calendar Record
X'21' - Begin Batch Utility Processing	X'92' - Add Calendar Record
X'22' - End Batch Utility Processing	X'93' - Update Calendar Record
X'31' - Begin System Task Processing	X'94' - Delete Calendar Record
X'32' - End System Task Processing	X'A1' - Read Event Record
X'41' - Begin Job Submit Processing	X'A2' - Add Event Record
X'42' - End Job Submit Processing	X'A3' - Post Event Record
X'71' - Open JCL Data Set	X'A4' - Reset Event Record
X'81' - Read Job Request Record	X'F1' - Convert Job Request Record

Parm 2 - Address of the PJSVT

The following parameter is passed for function codes X'41', X'42', X'81', X'82', X'83', X'84', and X'F1':

Parm 3 - Address of the PJSQ Job Request Record

The following parameter is passed for function codes X'91', X'92', X'93', and X'94':

Parm 3 - Address of the PJSQ Calendar Record

The following parameter is passed for function codes X'A1', X'A2', X'A3', and X'A4':

Parm 3 - Address of the PJSQ Event Record

The following parameters are passed for function code X'41':

Parm 3 - Address of 44-byte Data Set Name

Parm 4 - Address of 8-byte Member Name

Parm 5 - Address of 6-byte Volume Serial Number

For more information on the PJSQ Job Request record, the PJSQ Calendar record, the PJSQ Event record, and the PJS Vector Table formats, please refer to Appendix B.

4.3.2 PJSSECX Return Codes

On return, register 15 should be set to one of the following return codes:

0 Access Permitted

4 Access Denied

4.4 The PJS Installation Submit Exit (PJSSUBX)

PJSSUBX is an exit routine that enables the installation to examine and, if necessary, modify the submitted JCL or suppress the job. This exit will be called by the PJS each time a JCL image is to be passed to the internal reader.

The member ASMEXIT in the PJSINST library contains a sample job to link-edit PJSSUBX. The PJSSECX module must be link-edited with the RENT, AMODE=31, and RMODE=ANY attributes specified. The load module should be placed into the PJSLINK library.

The PJSSUBX exit is called using standard IBM linkage conventions. Register 15 will point to the program entry point address. The exit should return to the address passed in register 14. Register 13 will point to a 72-byte save area into which the exit should save the caller's registers. On return, the caller's registers (except register 15, which is set to the return code) must be restored.

4.4.1 PJSSUBX Input Parameters

Register 1 will point to the following parameter list:

- Parm 1 - Address of the PJSVT (PJS Vector Table)
- Parm 2 - Address of the PJS Job Request Record being processed
- Parm 3 - Address of the 80-byte JCL image last read by the system
- Parm 4 - Address of the 80-byte JCL image to be inserted by the exit

After the last JCL image has been passed to the internal reader, the exit will be called with Parm 3 equal to 0 (zero). This enables the exit to add any additional JCL statements required to the end of the job by generating a return code of 4.

The JCL image area pointed to by Parm 4 is initialized to blank characters when the exit is called. The JCL image placed in the area by the exit will be used only if the return code is set to 4.

For more information on the PJS Vector Table (PJSVT) and the PJS Request Queue Entry, which contains information pertaining to the Job Request Record, please refer to Appendix B.

4.4.2 PJSSUBX Return Codes

On return, register 15 should be set to one of the following return codes:

- 0 Pass the JCL image pointed to by Parm 3. If Parm 3 is non-zero, call the PJS Submit Exit again with the next JCL image; if Parm 3 is 0 (zero), the submission is complete and the exit will not be called again.
- 4 Insert the JCL image pointed to by Parm 4. Call the exit again with the same JCL image.
- 8 Delete the JCL image pointed to by Parm 3. If Parm 3 is non-zero, call the PJS Submit Exit again with the next JCL image; if Parm 3 is 0 (zero), the submission is complete and the exit will not be called again.
- 12 Suppress the job submission. A /*PURGE statement will be passed to the internal reader and the exit will not be called again.

4.4.3 How to Use the PJSSUBX Exit

The PJS Installation Submit Exit can modify the JCL being submitted by using the JCL image area passed by Parm 3 and Parm 4, and by setting the return code:

- To pass an input JCL image without change, set the return code to 0 (zero).
- To modify an input JCL image before submission, change the input JCL image in the area pointed to by Parm 3, then set the return code to 0 (zero).
- To insert a JCL image before the input JCL image, place the JCL image to be inserted into the area pointed to by Parm 4, then set the return code to 4.
- To delete the JCL image before the input JCL image, set the return code to 8.
- To suppress the job submission, set the return code to 12. A /*PURGE statement will be passed to the internal reader.

The PJS Vector Table (PJSVT) contains four fullwords labelled **PJSVT_USER n** . The exit can use these fullwords to hold information or point to storage areas between calls to the exit. The exit must perform cleanup before it sets the return code to 0 (zero) or 4 when Parm 3 is zero, i.e., the call after the final input JCL image; it must also perform cleanup when it sets the return code to 12 to purge the job.

For more information on the PJSVT, refer to the section on the PJS Vector Table in Appendix B.

5. PJS Security Issues

Some thought must be given to system security and integrity. Without proper precautions, it may be possible for an unauthorized user to use PJS to submit jobs with a different user-ID. Exit points and sample exits are provided to insure a completely secure environment. The specific steps required to provide adequate security will vary greatly from installation to installation, so detailed instructions cannot be given. Rather, this section is intended to give an overview of some of the things that should be considered. Although much of the discussion assumes a RACF environment, it should be apply to other security environments as well.

5.1 Securing the PJS Software

The first step to securing PJS is to secure the PJS software itself. All of the PJS libraries, in general, should be protected from unauthorized modifications just as you would protect any other production software system. The PJSLOAD library and the PJS Request Queue should be further secured. The programs contained in the PJSLOAD library are not generally required to be made available to users. The PJSLOAD library should therefore be in a data set with limited access.

The PJS system task (PJTASK) must be authorized so that it can send messages to online TSO users via the TSO SEND command. Also, many security schemes (including that provided by the sample security exit provided) will require authorization. The PJS Request Queue Maintenance Utility (PJSQMNT) requires authorization to call IEBCOPY to compress the PJS JCL Spool data set. No other programs require authorization, and should not be link-edited with AC=1.

5.2 Securing the PJS Data Sets

Most of the effort to secure PJS will involve protecting the PJS Request Queue and the PJS JCL Spool data sets from unauthorized access. This is complicated by the fact that the user must be able to update these data sets when using the PJS programs. The user must not, however, be able to update them by any other means. This can be accomplished using RACF program access control through a conditional access list.

To protect the data set with program access control, RACF profiles for the PROGRAM class must be created for the PJS TSO commands in the PJSCMD library (a generic profile name may be used for this purpose). The PROGRAM profile(s) used must include the PJSCMD and PJSLINE libraries in the ADDMEM clause with the NOPADCHK option. A RACF profile for the PJS request queue data set must be created with a conditional access list granting access when the PJS programs in PJSCMD are being executed. Member RACFPROF in the PJSINST library contains sample RACF commands for creating the RACF profiles. Please note that this is an example only. Some of the statements may not be appropriate for your installation, and other RACF statements may be required. Refer to the appropriate RACF documentation for authoritative information.

The PJS TSO commands (listed in Appendix A) may also need to be defined in the TSO authorized command table (IKJTSO00 in SYS1.PARMLIB, or CSECT IKJEFT2). (Although the PJS TSO commands do not require authorization, RACF program access control requires the secure environment TSO creates for a program in the authorized command list. Since the programs are not linked with AC=1, no APF authorization is actually conveyed to the program.) Member IKJTSO00 in the PJSINST library contains sample IKJTSO00 parameters to define the PJS TSO commands.

5.3 Securing the PJS Request Queue Records

The PJS Installation Security Exit can be used to define the type of access each user can have to each job request record in the PJS Request Queue. When you define access rules, remember that if a user can add or update job requests having an Owner-ID other than her or his own, that user may be able to submit jobs using the other user's User-ID. (This might depend on how you protect submitted batch jobs.) It is recommended that either users be permitted to add and update only job requests with their own User-ID, or that the PJS Installation Security Exit be written to record the RACF User-ID of the last user to update the request be recorded in the PJS Job Request record Installation Data Area, and use that User-ID for data set access and job submission.

5.4 Securing Batch Jobs Submitted by PJS

PJS submits batch jobs from the PJS System Task. The main task attaches a subtask to perform the actual submit. The submit subtask will terminate when the submit is complete. The PJS Installation Security exit is called at the beginning of the submit. The security exit can issue a RACINIT macro for the user owning the request record. This will update the submit task's TCB to point to the ACEE created by the RACINIT for the owning user. All JCL data set access will then be done under the user-ID of the job request owner. Alternatively, the PJS Installation Security Exit is also called when the JCL data set is opened.

The PJS system task submits the job exactly as it is in the JCL data set. The installation should take whatever steps are necessary to ensure that the job is run under the correct RACF User-ID and Group. In a JES2 environment, Exit 2 can be very useful for this purpose. Exit 2 has access to the RACF User-ID and Group of the task submitting the job. (This is in RIDSUSR and RIDSGRP in the JES2 DCT for the reader. The DCT address is in PCEDCT in the JES2 PCE, which is in register 13 at entry to the exit.) The exit can force these fields into the job card for every batch job submitted in the system. Once it can be certain that every job has the correct USERID and GROUP parameters on the JOB card, batch jobs can be run without further password verification. Other schemes may also be possible.

6. PJS Operator Commands

PJS operator commands are used by the system operator to start, stop, and control the activity of the PJS System Task. Operator commands are entered at the system console, or through any other facility for entering operator commands (such as SDSF). Specific details on how to enter operator commands can be found in *MVS System Commands*.

6.1 The START Command

The START command is used to start the PJS System Task. The command format is:

```
START  task-name  
S
```

6.2 The STOP Command

The STOP command is used to stop the PJS System Task. There may be a short delay between the time the command is entered and the time the system actually shuts down. It is important that the task be allowed to finish its processing. Do not use the CANCEL command unless absolutely necessary. The command format is:

```
STOP  task-name  
P
```

or

```
MODIFY task-name,STOP  
F
```

6.3 The SCAN Command

The SCAN command is used cause an immediate scan of the PJS Request Queue. This should rarely be necessary, but may be convenient from time to time. The command format is:

```
MODIFY task-name,SCAN  
F
```


7. PJS Utilities

The Personal Job Scheduler system provides several utilities to help install and maintain PJS. Only the Site Administrator, a systems programmer, the person who installs PJS, or a systems operator should use any of the PJS Utilities.

The following utilities will be discussed:

Utility Name	Utility Description (Relevant Section)
PJSQINIT	PJS Request Queue Initialization Utility (Section 7.1)
PJSQMNT	PJS Request Queue Maintenance Utility (Section 7.2)

The PJS Request Queue Initialization Utility (PJSQINIT) initializes a new PJS Request Queue data set.

The PJS Request Queue Maintenance Utility (PJSQMNT) cleans up the PJS Request Queue and the PJS JCL Spool.

7.1 The PJS Request Queue Initialization (PJSQINIT) Utility

PJSQINIT is a utility that initializes a new PJS Request Queue data set. In most cases, a PJS Request Queue is initialized only during PJS installation.

Before you run **PJSQINIT**, the PJS Request Queue must be defined and allocated with IDCAMS. The new PJS Request Queue data set must be defined with RECSZ(144,4084), KEYS(17,0), and SHR(2). Most other data set attributes can be changed to optimize VSAM processing. For most sites, 1 or 2 cylinders of space should be sufficient.

The PJS Request Queue is initialized by writing a Queue Control Record into an empty VSAM KSDS.

PJSQINIT has no input parameters. The following **DD** statements are required:

PJSQ is the PJS Request Queue data set.

SYSOUT is the message data set, usually **SYSOUT=***.

SYSABEND is the dump output data set, usually **SYSOUT=***.

The member DEFPJSQ in the PJSINST library provides a sample job to define and initialize the PJS Request Queue data set. This job should be tailored to fit your installation standards, paying particular attention to the job card, the PJS Request Queue data set name, the volume serial number, and the allocation quantity.

7.2 The PJS Request Queue Maintenance (PJSQMNT) Utility

PJSQMNT is a utility that performs several maintenance functions on the PJS Request Queue and the PJS JCL Spool. This utility should be run periodically to eliminate out-of-date records from these PJS data sets.

PJSQMNT performs the following functions:

1. Deletes old job requests.
2. Deletes unreferenced event records.
3. Deletes unreferenced owner records.
4. Deletes unreferenced PJS JCL Spool members.
5. Re-synchronizes PJS JCL Spool record counts.
6. Compresses the PJS JCL Spool data set.

Deletion of old job request records is controlled by the **CMPTIME**, **DISTIME**, and **ERRTIME** options in the PJS Options Table (PJSOPT). For more information on PJSOPT, please refer to Chapter 3.

PJSQMNT enqueues the PJS Request Queue while it runs: no other access is allowed when **PJSQMNT** runs. You can use PJS to schedule **PJSQMNT** to run at relatively quiet times of day. Since PJS is usually used to run jobs off-hours, it's a good idea to run this utility late in the morning or early in the afternoon.

PJSQMNT has no input parameters. The following **DD** statements are required:

SYSOUT is the message data set, usually **SYSOUT=***.

SYSABEND is the dump output data set, usually **SYSOUT=***.

The member PJSQMNT in the PJSINST library provides a sample job to run the PJS Request Queue Maintenance Utility. This job should be tailored to fit your installation standards, paying particular attention to the job card. It's a good idea back up the PJS Request Queue and the PJS JCL Spool before you run this utility.

Appendix A. PJS TSO Commands

The following table lists the TSO commands and the access level required to the PJS Request Queue and the PJS JCL Spool data sets. Several of these commands are used internally by the PJS/ISPF interface, and are not documented in the *PJS User Guide*.

Command Name	Program Name	PJSQ Access	JCL Spool Access	Description
PJREQADD PJRA PJADD PJA	PJSTJA	Update	Update	TSO Add Job Request
PJREQMOD PJRM PJMODIFY PJMOD PJM	PJSTJM	Update	Update	TSO Modify Job Request
PJREQDEL PJRD PJDELETE PJDEL PJD	PJSTJD	Update	Update	TSO Delete Job Request
PJREQLST PJRL PJLIST PJL	PJSTJL	Read	Read	TSO List Job Requests
PJCALADD PJCA	PJSTCA	Update	None	TSO Add Calendar
PJCALMOD PJCM	PJSTCM	Update	None	TSO Modify Calendar
PJCALDEL PJCD	PJSTCD	Update	None	TSO Delete Calendar
PJCALIST PJCL	PJSTCL	Read	None	TSO List Calendars
PJEVPOST PJEP	PJSTEP	Update	None	TSO Post Event
PJEVRSET PJER	PJSTER	Update	None	TSO Reset Event
PJEVLIST PJEL		Read	None	TSO List Events

Command Name	Program Name	PJSQ Access	JCL Spool Access	Description
PJSITCA	PJSITCA	Update	None	Command Processors used internally by the ISPF Interface
PJSITCD	PJSITCD	Update	None	
PJSITCG	PJSITCG	Read	None	
PJSITCJ	PJSITCJ	Read	None	
PJSITCL	PJSITCL	Read	None	
PJSITCM	PJSITCM	Update	None	
PJSITEG	PJSITEG	Read	None	
PJSITEJ	PJSITEJ	Read	None	
PJSITEL	PJSITEL	Read	None	
PJSITEP	PJSITEP	Update	None	
PJSITER	PJSITER	Update	None	
PJSITJA	PJSITJA	Update	Update	
PJSITJD	PJSITJD	Update	Update	
PJSITJG	PJSITJG	Read	None	
PJSITJJ	PJSITJJ	Read	Read	
PJSITJL	PJSITJL	Read	None	
PJSITJM	PJSITJM	Update	Update	

Appendix B. PJS Data Areas

The following sections contain information about the PJS Options Table, the PJS Vector Table, the PJS Request Queue Entry, the PJSQ Queue Control Record, the PJSQ Owner Record, the PJSQ Job Request Record, the PJSQ Calendar Record, and the PJSQ Event Record. This information may change from release to release.

B.1 The PJS Options Table

DSECT Name: PJSOPT
Macro ID: PJSOPT
Size: 292 Bytes
Function: The PJS Options Table contains installation-defined constants that are used throughout the PJS system.

Offsets		Type	Length	Name	Description
Dec	Hex				
0	(0)			PJSOPT	PJS Options Table.
0	(0)	CHARACTER	8	PJSOPT_TBL_ID	PJSOPT Table Identifier. =CL8'PJSOPT'
8	(8)	CHARACTER	8	PJSOPT_DATE	PJSOPT Generation Date.
16	(10)	CHARACTER	5	PJSOPT_TIME	PJSOPT Generation Time.
21	(15)	CHARACTER	8	PJSOPT_FMID	PJSOPT Generation Macro FMID.
29	(1D)	CHARACTER	8	PJSOPT_RMID	PJSOPT Generation Macro RMID.
37	(25)		3	RESERVED	Reserved for future use.
40	(28)		4	RESERVED	Reserved for future use.
44	(2C)	CHARACTER	44	PJSOPT_PJSQDSN	PJSQ Data Set Name.
88	(58)	CHARACTER	44	PJSOPT_JCLSDSN	JCL Spool Data Set Name.
132	(84)		48	RESERVED	Reserved for future use.
180	(B4)	CHARACTER	8	PJSOPT_TMPUNIT	Temporary data set unit name.
188	(BC)	CHARACTER	8	PJSOPT_ENQNAME	QNAME for serialization.
196	(C4)		8	RESERVED	Reserved for future use.
204	(CC)	BINARY	4	PJSOPT_SCANINT	PJSQ long scan interval time.
208	(D0)	BINARY	4	PJSOPT_RETRY_NUM	Job submit retry max number.
212	(D4)	BINARY	4	PJSOPT_RETRY_INT	Job submit retry interval.
216	(D8)	BINARY	4	PJSOPT_CMPTIME	Completed hold time.
220	(DC)	BINARY	4	PJSOPT_DISTIME	Disabled hold time.
224	(E0)	BINARY	4	PJSOPT_ERRTIME	Error hold time.
228	(E4)		4	RESERVED	Reserved for future use.
232	(E8)	BINARY	4	PJSOPT_JCLSLIM1	JCL Spool limit for request.
236	(EC)	BINARY	4	PJSOPT_JCLSLIM2	JCL Spool limit for owner.
240	(F0)	BINARY	4	PJSOPT_TMPBLK	Temporary data set blocksize.
244	(F4)	BINARY	4	PJSOPT_TMPPRIM	Temporary data set primary allocation.
248	(F8)	BINARY	4	PJSOPT_TMPSEC	Temporary data set secondary allocation.

Offsets		Type	Length	Name	Description
Dec	Hex				
252	(FC)		4	RESERVED	Reserved for future use.
256	(100)	BINARY	4	PJSOPT_ISPFQSZ	ISPF List Table size.
260	(104)	BINARY	4	PJSOPT_PTBLSZ	PJSTASK Post/Reset Table size.
264	(108)	BINARY	4	PJSOPT_RRTBLSZ	PJSTASK Ready Request Table size.
268	(10C)	BINARY	4	PJSOPT_OTBLSZ	PJSQMNT Owner Table size.
272	(110)	BINARY	4	PJSOPT_ETBLSZ	PJSQMNT Event Table size.
276	(114)	BINARY	4	PJSOPT_STBLSZ	PJSQMNT Spool Member Table size.
280	(118)		4	RESERVED	Reserved for future use.
284	(11C)	CHARACTER	1	PJSOPT_RDRCLS	INTRDR SYSOUT Class.
285	(11D)	BITSTRING	1	PJSOPT_FLAGS	Option Flags.
		1		PJSOPT_TSOAUTH	Call TSO commands from an authorized environment.
		. 1		PJSOPT_JCLSAVEN	JCL 'NOSAVE' may be used.
		. . 1		PJSOPT_JCLSAVES	JCL 'SAVE' may be used.
		. . . 1		PJSOPT_JCLSAVED	JCL 'SAVE' is default.
286	(11E)	BINARY	1	PJSOPT_DATEFMT	Date format
		X'00'		PJSOPT_DFMT_MDY	mm/dd/yyyy
		X'01'		PJSOPT_DFMT_DMY	dd/mm/yyyy
287	(11F)		5	RESERVED	Reserved for future use.
292	(124)			PJSOPT_LEN	Length of PJSOPT Table.

B.2 The PJS Vector Table

DSECT Name: PJSVT
 Macro ID: PJSVT
 Size: 560 Bytes
 Function: The PJS Vector Table contains run time global pointers and flags. The PJSVT is created by the main routine and then passed to each subroutine and subtask it calls.

Offsets		Type	Length	Name	Description
Dec	Hex				
0	(0)			PJSVT	PJS Vector Table.
0	(0)	CHARACTER	8	PJSVT_TBL_ID	PJSVT Table Identifier. =CL8'PJSVT'

Installation options. May be modified by the Installation Options Exit (PJSOPTX).

8	(8)	CHARACTER	44	PJSVT_PJSQDSN	PJSQ Data Set Name.
52	(34)	CHARACTER	44	PJSVT_JCLSDSN	JCL Spool Data Set Name.
96	(60)		48	RESERVED	Reserved for future use.
144	(90)	CHARACTER	8	PJSVT_TMPUNIT	Temporary data set unit name.
152	(98)	CHARACTER	8	PJSVT_ENQNAME	QNAME for serialization.
160	(A0)		8	RESERVED	Reserved for future use.
168	(A8)	BINARY	4	PJSVT_SCANINT	PJSQ long scan interval time.
172	(AC)	BINARY	4	PJSVT_RETRY_NUM	Job submit retry max number.
176	(B0)	BINARY	4	PJSVT_RETRY_INT	Job submit retry interval.
180	(B4)	BINARY	4	PJSVT_CMPTIME	Completed hold time.
184	(B8)	BINARY	4	PJSVT_DISTIME	Disabled hold time.
188	(BC)	BINARY	4	PJSVT_ERRTIME	Error hold time.
192	(C0)		4	RESERVED	Reserved for future use.
196	(C4)	BINARY	4	PJSVT_JCLSLIM1	JCL Spool limit for request.
200	(C8)	BINARY	4	PJSVT_JCLSLIM2	JCL Spool limit for owner.
204	(CC)	BINARY	4	PJSVT_TMPBLK	Temporary data set blocksize.
208	(D0)	BINARY	4	PJSVT_TMPPRIM	Temporary data set primary allocation.
212	(D4)	BINARY	4	PJSVT_TMPSEC	Temporary data set secondary allocation.
216	(D8)		4	RESERVED	Reserved for future use.
220	(DC)	BINARY	4	PJSVT_ISPFQSZ	ISPF List Table size.
224	(E0)	BINARY	4	PJSVT_PTBLSZ	PJTASK Post/Reset Table size.
228	(E4)	BINARY	4	PJSVT_RRTBLSZ	PJTASK Ready Request Table size.
232	(E8)	BINARY	4	PJSVT_OTBLSZ	PJSQMNT Owner Table size.
236	(EC)	BINARY	4	PJSVT_ETBLSZ	PJSQMNT Event Table size.
240	(F0)	BINARY	4	PJSVT_STBLSZ	PJSQMNT Spool Member Table size.
244	(F4)		4	RESERVED	Reserved for future use.
248	(F8)	CHARACTER	1	PJSVT_RDRCLS	INTRDR SYSOUT Class.

Offsets		Type	Length	Name	Description
Dec	Hex				
249	(F9)	BITSTRING 1	1	PJSVT_OPT_FLAGS PJSVT_OPT_TAUTH	Option Flags. Call TSO commands from an authorized environment.
		. 1		PJSVT_OPT_JSAVEN	JCL 'NOSAVE' may be used.
		. . 1		PJSVT_OPT_JSAVES	JCL 'SAVE' may be used.
		. . . 1		PJSVT_OPT_JSAVED	JCL 'SAVE' is default.
250	(FA)	BINARY X'00'	1	PJSVT_DATEFMT PJSVT_DFMT_MDY PJSVT_DFMT_DMY	Date format mm/dd/yyyy dd/mm/yyyy
251	(FB)		5	RESERVED	Reserved for future use.

Installation data areas. May be used by installation exits.

256	(100)	BINARY	4	PJSVT_USER1	Installation use.
260	(104)	BINARY	4	PJSVT_USER2	Installation use.
264	(108)	BINARY	4	PJSVT_USER3	Installation use.
268	(10C)	BINARY	4	PJSVT_USER4	Installation use.

System environment data areas. Not to be modified by installation exits.

272	(110)	CHARACTER	8	PJSVT_LEVEL	PJS Version/Release/Level.
280	(118)	CHARACTER	8	PJSVT_PGMNAME	Main program name.
288	(120)	CHARACTER C'I C'J' C'S' C'T'	1	PJSVT_ENV PJSVT_ENV_ISP PJSVT_ENV_JOB PJSVT_ENV_STC PJSVT_ENV_TSO	System environment indicator ISPF interface. Batch utility job. PJS System Task. TSO command processor.
289	(121)	BITSTRING 1	1	PJSVT_SYS_FLAGS PJSVT_SYS_APF	System flags. APF authorized environment.
290	(122)		2	RESERVED	Reserved for future use.
292	(124)	ADDRESS	4	PJSVT_CPPL	Pointer to TSO CPPL.
296	(128)		8	RESERVED	Reserved for future use.

Pointers to routines. Not to be modified by installation exits.

304	(130)	ADDRESS	4	PJSVT_PJSALLOC	Pointer to Dynamic Allocation Service Routine.
308	(134)	ADDRESS	4	PJSVT_PJSCALFM	Pointer to Calendar Format Service Routine.
312	(138)	ADDRESS	4	PJSVT_PJSCALID	Pointer to Calendar-ID Input Service Routine.
316	(13C)	ADDRESS	4	PJSVT_PJSCALU	Pointer to Calendar Utility Service Routine.

Offsets		Type	Length	Name	Description
Dec	Hex				
320	(140)	ADDRESS	4	PJSVT_PJSDATIM	Pointer to Date/Time Service Routine.
324	(144)	ADDRESS	4	PJSVT_PJSESTAE	Pointer to ESTAE Routine.
328	(148)	ADDRESS	4	PJSVT_PJSEVNFM	Pointer to Event Format Service Routine.
332	(14C)	ADDRESS	4	PJSVT_PJSEVNID	Pointer to Event-ID Input Service Routine.
336	(150)	ADDRESS	4	PJSVT_PJSFREQ	Pointer to Calculate Next Run Date/Time Service Routine.
340	(154)	ADDRESS	4	PJSVT_PJSJIO	Pointer to JCL Data Set I/O Service Routine.
344	(158)	ADDRESS	4	PJSVT_PJSJRQFM	Pointer to Job Request Format Service Routine.
348	(15C)	ADDRESS	4	PJSVT_PJSJRQID	Pointer to Job Request-ID Input Service Routine.
352	(160)	ADDRESS	4	PJSVT_PJSMMSG	Pointer to Message Format Service Routine.
356	(164)	ADDRESS	4	PJSVT_PJSMMSGC	Pointer to Console Message Service Routine.
360	(168)	ADDRESS	4	PJSVT_PJSMMSGI	Pointer to ISPF Message Service Routine.
364	(16C)	ADDRESS	4	PJSVT_PJSMMSGP	Pointer to Print Message Service Routine.
368	(170)	ADDRESS	4	PJSVT_PJSMMSGT	Pointer to TSO Terminal Message Service Routine.
372	(174)	ADDRESS	4	PJSVT_PJSMMSGU	Pointer to TSO Send Message Service Routine.
376	(178)	ADDRESS	4	PJSVT_PJSQIO	Pointer to PJSQ I/O Service Routine.
380	(17C)	ADDRESS	4	PJSVT_PJSSIO	Pointer to JCL Spool I/O Service Routine.
384	(180)	ADDRESS	4	PJSVT_PJSSPOOL	Pointer to Copy JCL to Spool Service Routine.
388	(184)	ADDRESS	4	PJSVT_PJSTEMP	Pointer to Copy JCL to Temporary Data Set Service Routine.
392	(188)		16	RESERVED	Reserved for future use.
408	(198)	ADDRESS	4	PJSVT_PJSIDFX	Pointer to Installation Data Exit Routine.
412	(19C)	ADDRESS	4	PJSVT_PJSSECX	Pointer to Installation Security Exit.
416	(1A0)	ADDRESS	4	PJSVT_PJSSUBX	Pointer to Installation Submit Exit.
420	(1A4)		8	RESERVED	Reserved for future use.
428	(1AC)	ADDRESS	4	PJSVT_ISPLINK	Pointer to ISPF Service Routine.
432	(1B0)		8	RESERVED	Reserved for future use.
Execution time parameters. Not to be modified by installation exits.					
440	(1B8)	ADDRESS	4	PJSVT_MSGDEST	Pointer to default message destination list.

Offsets		Type	Length	Name	Description
Dec	Hex				
444	(1BC)	BINARY	4	PJSVT_TEST_DATE	Used for testing.
448	(1C0)	BINARY	4	PJSVT_TEST_TIME	Used for testing.
452	(1C4)		4	RESERVED	Reserved for future use.
PJSMGRP data areas. Not to be modified by installation exits.					
456	(1C8)	ADDRESS	4	PJSVT_PRTDCB	Pointer to printer DCB.
460	(1CC)	BINARY	4	PJSVT_PRTDCB_LN	Length of printer DCB.
464	(1D0)	BITSTRING	1	PJSVT_PRT_FLAGS	Printer file status flags.
		1... ..		PJSVT_PRT_OPEN	Printer data set open.
		.1... ..		PJSVT_PRT_CLOSED	Printer data set closed.
465	(1D1)		7	RESERVED	Reserved for future use.
PJSQIO data areas. Not to be modified by installation exits.					
472	(1D8)	ADDRESS	4	PJSVT_PJSQIOD	Pointer to PJSQIO Data Area.
476	(1DC)	ADDRESS	4	PJSVT_PJSQIOD_LN	Length of PJSQIO Data Area.
480	(1E0)	ADDRESS	4	PJSVT_PQCTL_REC	Pointer to PJSQ Queue Control record.
484	(1E4)	ADDRESS	4	PJSVT_PQOWN_REC	Pointer to PJSQ Owner record.
488	(1E8)	ADDRESS	4	PJSVT_PQJRQ_REC	Pointer to PJSQ Job Request record.
492	(1EC)	ADDRESS	4	PJSVT_PQCAL_REC	Pointer to PJSQ Calendar record.
496	(1F0)	ADDRESS	4	PJSVT_PQEVN_REC	Pointer to PJSQ Event record.
500	(1F4)	BITSTRING	1	PJSVT_PJSQ_FLAGS	PJSQ file status flags.
		1... ..		PJSVT_PJSQ_ALLOC	PJSQ data set allocated.
		.1... ..		PJSVT_PJSQ_OPEN	PJSQ data set open.
501	(1F5)		11	RESERVED	Reserved for future use.
PJSSIO data areas. Not to be modified by installation exits.					
512	(200)	ADDRESS	4	PJSVT_PJSSIOD	Pointer to PJSSIO Data Area.
516	(204)	ADDRESS	4	PJSVT_PJSSIOD_LN	Length of PJSSIO Data Area.
520	(208)	BITSTRING	1	PJSVT_SPL_FLAGS	JCL Spool file status flags.
		1... ..		PJSVT_SPL_ALLOC	JCL Spool allocated.
		.1... ..		PJSVT_SPL_OPENI	JCL Spool open for input.
		..1.		PJSVT_SPL_OPENO	JCL Spool open for output.
		...1		PJSVT_SPL_OPENI	JCL Spool directory open.
521	(209)		7	RESERVED	Reserved for future use.

Offsets		Type	Length	Name	Description
Dec	Hex				
PJSJIO data areas. Not to be modified by installation exits.					
528	(210)	ADDRESS	4	PJSVT_PJSJIOD	Pointer to PJSJIO Data Area.
532	(214)	ADDRESS	4	PJSVT_PJSJIOD_LN	Length of PJSJIO Data Area.
536	(218)	BITSTRING	1	PJSVT_JCL_FLAGS	JCL Data Set file status flags.
		1		PJSVT_JCL_ALLOC	JCL Data Set allocated.
		. 1		PJSVT_JCL_OPEN	JCL Data Set open.
537	(219)		7	RESERVED	Reserved for future use.
PJSSUB data areas. Not to be modified by installation exits.					
544	(220)		8	RESERVED	Reserved for future use.
552	(228)	BITSTRING	1	PJSVT_RDR_FLAGS	INTRDR file status flags.
		1		PJSVT_RDR_ALLOC	INTRDR allocated.
		. 1		PJSVT_RDR_OPEN	INTRDR open.
553	(229)		7	RESERVED	Reserved for future use.
560	(230)			PJSVT_LEN	Length of PJS Vector Table.

B.3 The PJS Request Queue Entry

DSECT Name: PJSQE
 Macro ID: PJSQE
 Size: Variable
 Function: This DSECT describes the format of the PJS Request Queue Entry. This DSECT is used to map a general PJSQ record without the specific record type fields. This basically consists of the record length and the key. The data portion is referred to only by a single data area. The fields for each record type are described by other DSECTs.

Offsets		Type	Length	Name	Description
Dec	Hex				
0	(0)			PJSQE	PJS Request Queue Entry.

The record descriptor word does not appear in the VSAM data set.

0	(0)	BINARY	2	PJSQE_RECLEN	Record length (including this field).
2	(2)		2	RESERVED	Reserved.

The following fields define the record key.

4	(4)	GROUP	17	PJSQE_KEY	Record Key.
4	(4)	BINARY	1	PJSQE_TYPE	Record Type.
		X'01'		PJSQE_TYPE_CNTL	Queue Control Record.
		X'02'		PJSQE_TYPE_OWNER	Owner Record.
		X'03'		PJSQE_TYPE_JOB	Job Request Record.
		X'04'		PJSQE_TYPE_CAL	Calendar Record.
		X'05'		PJSQE_TYPE_EVNT	Event Record.
5	(5)	CHARACTER	8	PJSQE_OWNERID	Owner-ID of this record.
13	(D)	CHARACTER	8	PJSQE_RECID	Record-ID of this record.

The following fields define the record type dependant area. This area is redefined for each record type.

21	(15)		3	RESERVED	Reserved.
24	(18)	GROUP	4064	PJSQE_DATA	Record type dependant data area.
4088	(FF8)			PJSQE_LEN	Maximum length of PJSQ record.

B.4 The PJSQ Queue Control Record

DSECT Name: PJSQCTL
 Macro ID: PJSQE
 Size: 29 bytes
 Function: This DSECT describes the format of the PJSQ Queue Control Record. There is one Queue Control record for the PJS Request Queue. It contains global information of concern to the entire Request Queue.

Offsets		Type	Length	Name	Description
Dec	Hex				
0	(0)			PJSQCTL	PJSQ Queue Control Record.

The record descriptor word does not appear in the VSAM data set.

0	(0)	BINARY	2	PQCTL_RECLEN	Record length (including this field).
2	(2)		2	RESERVED	Reserved.

The following fields define the record key.

4	(4)	GROUP	17	PQCTL_KEY	Record Key.
4	(4)	BINARY	1	PQCTL_TYPE	Record Type (Always X'01').
5	(5)	CHARACTER	8	PQCTL_OWNERID	Owner-ID (Always X'00').
13	(D)	CHARACTER	8	PQCTL_RECID	Record-ID (Always X'00').

The following fields define the record type dependant area.

21	(15)		3	RESERVED	Reserved.
24	(18)	BINARY	4	PQCTL_NXT_JCLNUM	Next JCL Spool member number.
28	(1C)	BITSTRING	1	PQCTL_UPD_FLAGS	Update flags.
		1		PQCTL_UPD_JRQ	Job Request has been updated.
		. 1		PQCTL_UPD_CAL	Calendar has been updated.
		. . 1		PQCTL_UPD_EVNT	Event has been updated.
29	(1D)			PQCTL_LEN	Length of PJSQ Queue Control record.

B.5 The PJSQ Owner Record

DSECT Name: PJSQOWN
 Macro ID: PJSQE
 Size: 44 bytes
 Function: This DSECT describes the format of the PJSQ Owner Record. It contains information for each Owner-ID with records stored in the PJS Request Queue.

Offsets		Type	Length	Name	Description
Dec	Hex				
0	(0)			PJSQOWN	PJSQ Owner Record.

The record descriptor word does not appear in the VSAM data set.

0	(0)	BINARY	2	PQOWN_RECLEN	Record length (including this field).
2	(2)		2	RESERVED	Reserved.

The following fields define the record key.

4	(4)	GROUP	17	PQOWN_KEY	Record Key.
4	(4)	BINARY	1	PQOWN_TYPE	Record Type (Always X'02').
5	(5)	CHARACTER	8	PQOWN_OWNERID	Owner-ID.
13	(D)	CHARACTER	8	PQOWN_RECID	Record-ID (Always X'00').

The following fields define the record type dependant area.

21	(15)		3	RESERVED	Reserved.
24	(18)	BINARY	4	PQOWN_USE_DATE	Date of last Owner update.
28	(1C)	BINARY	4	PQOWN_USE_TIME	Time of last Owner update.
32	(20)	BINARY	4	PQOWN_SPOOLNUM	Number of JCL records on spool.
36	(24)	BINARY	2	PQOWN_NXT_REQNUM	Next Request Number to assign.
38	(26)		6	RESERVED	Reserved.
44	(2C)			PQOWN_LEN	Length of PJSQ Owner record.

B.6 The PJSQ Job Request Record

DSECT Names: PJSQJRQ
PJSQJRQC
PJSQJRQE
PJSQJRQI

Macro ID: PJSQE

Size: variable, 168 bytes minimum, 1884 bytes maximum.

Function: These DSECTs describe the format of the PJSQ Job Request Record. It contains information for each Job Request entered by the users. The record consists of a basic record area, followed by one or more variable length segments. A Calendar segment is present if calendar scheduling is used, an Event segment is present if event scheduling is used, and an Installation Data Area is present if it was created by the Installation Security Exit (PJSSECX). DSECT PJSQJRQ describes the format of the basic Job Request Record. DSECT PJSQJRQC describes the format of the Calendar Segment. DSECT PJSQJRQE describes the format of the Event Segment.

Offsets		Type	Length	Name	Description
Dec	Hex				
0	(0)			PJSQJRQ	PJSQ Job Request Record.

The record descriptor word does not appear in the VSAM data set.

0	(0)	BINARY	2	PQJRQ_RECLEN	Record length (including this field).
2	(2)		2	RESERVED	Reserved.

The following fields define the record key.

4	(4)	GROUP	17	PQJRQ_KEY	Record Key.
4	(4)	BINARY	1	PQJRQ_TYPE	Record Type (Always X'03').
5	(5)	CHARACTER	8	PQJRQ_OWNERID	Owner-ID.
13	(D)	CHARACTER	8	PQJRQ_REQNUM	Request Number.

The following fields define the record type dependant area.

21	(15)		3	RESERVED	Reserved.
24	(18)	BINARY	2	PQJRQ_CAL_OFF	Offset to Calendar Table.
26	(1A)	BINARY	2	PQJRQ_CAL_NUM	Number of Calendar Table entries.
28	(1C)	BINARY	2	PQJRQ_EVNT_OFF	Offset to Event Table.
30	(1E)	BINARY	2	PQJRQ_EVNT_NUM	Number of Event Table entries.
32	(20)	BINARY	2	PQJRQ_INST_OFF	Offset to Installation Data Area.
34	(22)	BINARY	2	PQJRQ_INST_LEN	Length of Installation Data Area.
36	(24)	BINARY	1	PQJRQ_STATUS	Job Request processing status.
		X'01'		PQJRQ_STAT_WAIT	Waiting for next run time.
		X'02'		PQJRQ_STAT_SUB	Currently in submit processing.
		X'03'		PQJRQ_STAT_ERR	Error encountered.
		X'04'		PQJRQ_STAT_CMPL	Completed.
		X'05'		PQJRQ_STAT_DIS	Disabled.

Offsets		Type	Length	Name	Description
Dec	Hex				
37	(25)	BINARY X'00' X'01' X'02' X'03'	1	PQJRQ_JSAVE PQJRQ_JSAVE_NO PQJRQ_JSAVE_YES PQJRQ_JSAVE_REFR PQJRQ_JSAVE_DEL	JCL Save option. JCL is not saved in Spool. JCL is saved in Spool. JCL refresh pending. JCL delete pending.
38	(26)		10	RESERVED	Reserved.
48	(30)	CHARACTER	44	PQJRQ_DSNAME	JCL Data Set Name.
92	(5C)	CHARACTER	8	PQJRQ_MEMBER	JCL Member Name.
100	(64)	CHARACTER	8	PQJRQ_SPOOLMEM	JCL Spool Member Name.
108	(6C)	BINARY	4	PQJRQ_SPOOLNUM	Number of JCL records on spool.
112	(70)	BINARY	4	PQJRQ_WNDW_TIME	Submit Window Time.
116	(74)	BINARY X'01' X'02' X'03'	1	PQJRQ_WNDW_OPT PQJRQ_WNDW_DIS PQJRQ_WNDW_ERR PQJRQ_WNDW_SKIP	Submit Window Failure Option. Place in DISABLED status. Place in ERROR status. Reschedule for next period.
117	(75)		1	RESERVED	Reserved.
118	(76)	BINARY	2	PQJRQ_RETRY_NUM	Number of submit retries.
120	(78)	BINARY	4	PQJRQ_LAST_DATE	Last Submit Date.
124	(7C)	BINARY	4	PQJRQ_LAST_TIME	Last Submit Time.
128	(80)	BINARY	4	PQJRQ_NEXT_DATE	Next Submit Date.
132	(84)	BINARY	4	PQJRQ_NEXT_TIME	Next Submit Time.
136	(88)	BINARY	4	PQJRQ_START_DATE	Starting Date.
140	(8C)	BINARY	4	PQJRQ_START_TIME	Starting Time.
144	(90)	BINARY	4	PQJRQ_END_DATE	Ending Date.
148	(94)	BINARY	4	PQJRQ_END_TIME	Ending Time.
152	(98)	BINARY X'01' X'02' X'03' X'04' X'05'	1	PQJRQ_FREQ PQJRQ_FREQ_ONCE PQJRQ_FREQ_PRD PQJRQ_FREQ_WKDAY PQJRQ_FREQ_EOM PQJRQ_FREQ_CAL	Job Submit Frequency Type. Once. Periodic. Day-of-Week. End-of-Month. Calendar.
153	(99)	BINARY X'01' X'02' X'03' X'04' X'05' X'06'	1	PQJRQ_PRD_UNIT PQJRQ_PRD_YRS PQJRQ_PRD_MOS PQJRQ_PRD_WKS PQJRQ_PRD_DAYS PQJRQ_PRD_HRS PQJRQ_PRD_MINS	Periodic Units. Years. Months. Weeks. Days. Hours. Minutes.
154	(9A)	BINARY	2	PQJRQ_PRD_QTY	Periodic Quantity.

Offsets		Type	Length	Name	Description
Dec	Hex				
156	(9C)	BITSTRING	1	PQJRQ_WKDAY	Day-of-Week to submit job.
		1... ..		PQJRQ_WKDAY_SUN	Sunday.
		.1... ..		PQJRQ_WKDAY_MON	Monday.
		..1.		PQJRQ_WKDAY_TUE	Tuesday.
		...1		PQJRQ_WKDAY_WED	Wednesday.
	 1...		PQJRQ_WKDAY_THU	Thursday.
	1..		PQJRQ_WKDAY_FRI	Friday.
	1.		PQJRQ_WKDAY_SAT	Saturday.
157	(9D)	BINARY	1	PQJRQ_EOM_DAYS	Days before End-of-Month.
158	(9E)		10	RESERVED	Reserved.
168	(A8)	GROUP	3920	PQJRQ_SEGMENTS	Record segment area.
4088	(FF8)			PQJRQ_LEN	Maximum length of PJSQ Job Request record.

The following fields define the Calendar Segment. The Calendar Segment is a table each entry of which contains information on a calendar referenced by the Job Request.

0	(0)			PJSQJRC	Calendar Table Segment.
0	(0)	GROUP	17	PQJRQ_CAL_ID	Calendar-ID.
0	(0)	BINARY	1	PQJRQ_CAL_TYPE	Calendar Type (Always X'04').
1	(1)	CHARACTER	8	PQJRQ_CAL_OWNID	Calendar Owner-ID.
9	(9)	CHARACTER	8	PQJRQ_CAL_NAME	Calendar Name.
17	(11)		3	RESERVED	Reserved.
20	(14)			PQJRQ_CAL_ELEN	Length of Calendar Table Entry.
		3		PQJRQ_CAL_MAX	Maximum number of Calendar Table Entries.

The following fields define the Event Segment. The Event Segment is a table each entry of which contains information on an event referenced by the Job Request.

0	(0)			PJSQJQE	Event Table Segment.
0	(0)	GROUP	17	PQJRQ_EVNT_ID	Event-ID.
0	(0)	BINARY	1	PQJRQ_EVNT_TYPE	Event Type (Always X'05').
1	(1)	CHARACTER	8	PQJRQ_EVNT_OWNID	Event Owner-ID.
9	(9)	CHARACTER	8	PQJRQ_EVNT_NAME	Event Name.
17	(11)	BITSTRING	1	PQJRQ_EVNT_FLAG	Job Request Event flags.
		1... ..		PQJRQ_EVNT_PREP	Event may be 'Preposted'.
18	(12)		2	RESERVED	Reserved.
20	(14)	BINARY	4	PQJRQ_EVNT_DATE	Date Event Posted.
24	(18)	BINARY	4	PQJRQ_EVNT_TIME	Time Event Posted.

Offsets		Type	Length	Name	Description
Dec	Hex				
28	(1C)	50		PQJRQ_EVNT_ELEN PQJRQ_EVNT_MAX	Length of Event Table Entry. Maximum number of Event Table Entries.

The following fields define the Installation Data Area. The Installation Data Area is a variable length area defined by the installation and set by the Installation Security Exit (PJSSECX).

0	(0)			PJSQJRQI	Installation Data Area.
0	(0)	CHARACTER	256	PQJRQ_INST_DATA	Installation Data Area.

B.7 The PJSQ Calendar Record

DSECT Name: PJSQCAL
 Macro ID: PJSQE
 Size: variable, 32 bytes minimum, 4088 bytes maximum.
 Function: This DSECT describes the format of the PJSQ Calendar Record. It contains a variable length bitstring of selected dates for the calendar.

Offsets		Type	Length	Name	Description
Dec	Hex				
0	(0)			PJSQCAL	PJSQ Calendar Record.

The record descriptor word does not appear in the VSAM data set.

0	(0)	BINARY	2	PQCAL_RECLEN	Record length (including this field).
2	(2)		2	RESERVED	Reserved.

The following fields define the record key.

4	(4)	GROUP	17	PQCAL_KEY	Record Key.
4	(4)	BINARY	1	PQCAL_TYPE	Record Type (Always X'04').
5	(5)	CHARACTER	8	PQCAL_OWNERID	Owner-ID.
13	(D)	CHARACTER	8	PQCAL_CALNAME	Calendar Name.

The following fields define the record type dependant area.

21	(15)		3	RESERVED	Reserved.
24	(18)	BINARY	4	PQCAL_START_DATE	Date on which calendar starts.
28	(1C)	BITSTRING 1	1	PQCAL_FLAGS PQCAL_UPDATE	Update flags. Calendar has been updated.
29	(1D)		1	RESERVED	Reserved.
30	(1E)	BINARY	2	PQCAL_DATES_LEN	Length of Calendar Dates bit string.
32	(20)	BITSTRING	4056	PQCAL_DATES	Calendar Dates. Each bit is one day, starting with the Start Date.
4088	(FF8)			PQCAL_LEN	Maximum length of PJSQ Calendar record.

B.8 The PJSQ Event Record

DSECT Name: PJSQEVN
 Macro ID: PJSQE
 Size: 33 bytes
 Function: This DSECT describes the format of the PJSQ Event Record. It contains a flag that indicates when the Job Request Events are to be posted or reset.

Offsets		Type	Length	Name	Description
Dec	Hex				
0	(0)			PJSQEVN	PJSQ Event Record.

The record descriptor word does not appear in the VSAM data set.

0	(0)	BINARY	2	PQEVN_RECLEN	Record length (including this field).
2	(2)		2	RESERVED	Reserved.

The following fields define the record key.

4	(4)	GROUP	17	PQEVN_KEY	Record Key.
4	(4)	BINARY	1	PQEVN_TYPE	Record Type (Always X'05').
5	(5)	CHARACTER	8	PQEVN_OWNERID	Owner-ID.
13	(D)	CHARACTER	8	PQEVN_EVNTNAME	Event Name.

The following fields define the record type dependant area.

21	(15)		3	RESERVED	Reserved.
24	(18)	BINARY	4	PQEVN_POST_DATE	Date event last posted.
28	(1C)	BINARY	4	PQEVN_POST_TIME	Time event last posted.
32	(20)	BITSTRING	1	PQEVN_FLAGS	Update flags.
		1		PQEVN_JOB_POST	Job Request Event post pending.
		. 1		PQEVN_JOB_RESET	Job Request Event reset pending.
33	(21)			PQEVN_LEN	Length of PJSQ Event record.

Appendix C. Summary of Changes

C.1 Changes for PJS Release 2.1.4

Functional Changes -

- The CPU authorization checking has been removed.
- Commands to display copyright and license information have been added to the ISPF menu.

Installation Changes -

- PJS is now distributed as a single file that can be distributed as part of a public software distribution tape, such as the MVS CBT Tape, or downloaded from the internet.
- PJS is now installed with full source code. All macros required for reassembling PJS (other than IBM macros that are shipped with MVS) are included.
- The sample installation JCL has been moved from the SMP/E JCLIN PDS to a separate PDS.
- The PJS documentation is now shipped as a zip file that can be downloaded to a workstation and expanded. The documentation is shipped in Microsoft Word, Adobe Acrobat PDF, and HTML files.
- The PJS Request Queue Conversion Utility (PJSQCONV) has been removed.

C.2 Changes for PJS® Release 2.1.3

Functional Changes -

- The default century for 2-digit years is changed from 19xx to 20xx.

Installation Changes -

- PJS Documentation is now included on the installation tape in PDF (Adobe Acrobat Reader) format.

Additional PJS R2.1 PTF's and APAR's Incorporated -

PPJ0029 - ISPF Help for Delete Job Request panel (**PJSPJDEL**) gives "Panel not found" dialog error.

PPJ0030 - PJS Security Exit (**PJSSECX**) not called from **PJSTASK** for "Begin System Task" and "End System Task".

PPJ0031 - Documentation for sample PJS Installation Security Exits (**PJSSECX**) is incorrect.

OPJ0034 - List Events for Owner does not list Events when the corresponding Owner record does not exist.

PPJ0035 - Scope of enqueue on PJS Request Queue changed from **SYSTEM** to **SYSTEMS**.

PPJ0036 - Scope of dequeue on PJS Request Queue changed from **SYSTEM** to **SYSTEMS**.

C.3 Changes for PJS® Release 2.1

Functional Changes -

- PJS can process dates in the *dd/mm/yyyy* format (commonly used in Europe), in addition to the *mm/dd/yyyy* format (commonly used in the U.S.). The format used is controlled by the new **DATEFMT** parameter of the **PJSOPT** macro.
- PJS ISPF commands may be abbreviated to 2 characters.
- A PJS Tutorial Index is provided.
- The name of the Batch Event Post Utility is changed from '**PJSEVENT**' to '**PJSPOST**'. The old name, '**PJSEVENT**', is still supported as an alias of '**PJSPOST**'.
- A new Batch Event Reset Utility (**PJSRESET**) is available. This utility complements the Batch Event Post Utility (**PJSPOST**).
- The Batch Event Post Utility (**PJSPOST**) and the Batch Event Reset Utility (**PJSRESET**) are described in the PJS ISPF Tutorial.

Installation Changes -

- A User Contributed Routines library is provided on file 11 of the installation tape. These routines are provided by PJS users and distributed on an 'as-is' basis. They are not supported by NNS Information Services, nor have they been tested by NNS. Use of these routines is at the customers own risk.
- The installation options are checked for validity at program initialization time. In addition, the PJS System Task will display the current option values.
- The default for the **TSOAUTH** parameter of the **PJSOPT** macro is changed from '**NO**' to '**YES**'. '**NO**' is rarely needed, while '**YES**' is frequently required.
- The PJS Request Queue Maintenance Utility (**PJSQMNT**) will treat a zero value for the **CMPTIME**, **DISTIME**, or **ERRTIME** options as indicating 'no limit'.
- A sample PJS Options Exit (**PJSOPTX1**) is provided that will allow an installation to run multiple copies of PJS on a single system.
(PJS 2.0 PTF PPJ0017)

- The internal formats of the PJS Options Table (**PJSOPT**) and the PJS Vector Table (**PJSVT**) have changed. The PJS Options Table and any user exits must be reassembled.

Problems Fixed -

- When the PJS System Task attempted to submit a job, and the JCL data set was in use by another task, the job request was failed with message **PJS401E**, and the job request was placed in the **'ERROR'** status.

The PJS System Task is changed to requeue the job request and retry the submission at a later time. The maximum number of attempts and the interval between each attempt is controlled by the new **RETRY** parameter of the **PJSOPT** macro.

- When the PJS System Task attempted to submit a job, and no internal readers were available, the job request was failed with message **PJS401E**, and the job request was placed in the **'ERROR'** status.

The PJS System Task is changed to wait until an internal reader is available. There is no limit to how long PJS will wait.

- After the PJS TSO commands (including those used by the PJS ISPF interface) terminated, the PJS Request Queue data set remained allocated.

The PJS Request Queue data set is automatically deallocated when the TSO commands terminate. The PJS System Task will continue to leave the PJS Request Queue data set allocated.

- When the PJS Queue Maintenance Utility (**PJSQMNT**) was run, and the PJS JCL Spool was not used by the installation, message **PJS401E** and abend **2103** resulted.

PJSQMNT determines if the PJS JCL Spool is in use by testing **PJSVT_JCLSDSN** for nulls, but the **PJSOPT** macro sets the PJS JCL Spool Data Set Name to spaces. **PJSQMNT** is changed to check the **PJSVT_OPT_JSAVES** bit of **PJSVT_OPT_FLAGS** (set by the **JCLSAVE** option), instead of **PJSVT_JCLSDSN** to determine if the JCL Spool data set is in use.
(Supercedes PJS 2.0 PTF PPJ0023)

- The PJS System Task did not correctly check the Ready Request Table Entry against the current system time to determine if a request was ready to be submitted. Also, after a job request was successfully processed, its Ready Request Table Entry was not updated. The result was that every minute **PJSTASKJ** was called to process each job request in the Ready Request Table. Although **PJSTASKJ** performed its own check, the result was a great deal of unnecessary activity against the PJS Request Queue.

The PJS System Task is fixed to correctly check the Ready Request Table Entry against the current system time. Also, **PJSTASKJ** is fixed to update the Ready Request Table Entry to reflect the new submit date (if any) of the job request.

- The use of attributes in the PJS ISPF panels was not always consistent. Also, for some panels, the use of the 'skip' attribute after some input fields is desirable.

The use of attributes in the PJS ISPF panels has been standardized. Input fields with no existing value will be filled with underscores ('_'). The attribute terminating most input fields will have the 'noskip' attribute. Some selection fields will have 'skip', where this facilitates data entry.

- When the PJS Tutorial was entered using the 'T' option on the PJS Main Menu panel, at the end of the tutorial, the IBM ISPF tutorial was entered, instead of returning to the PJS Tutorial main panel.

The PJS ISPF Interface is changed to correctly set the ISPF Tutorial control variables **ZHTOP** and **ZHINDEX** in the shared variable pool.

Additional PJS R2.0 PTF's Incorporated -

PPJ0018 - Message **PJS749I** incorrectly displayed by PJS Queue Maintenance Utility (**PJSQMNT**).

PPJ0020 - ISPF List panels do not display national characters (\$, #, or @) in dynamic area.

PPJ0021 - ISPF Copy Calendar function does not set the new Calendar-ID.

PPJ0022 - ISPF List panels are being formatted with nulls (X'00') imbedded in the lines.

PPJ0024 - Message **PJS234E** issued by PJS System Task with sample security exit **PJSSECX** or **PJSSECX2**.

PPJ0025 - Abend **U2706** issued by PJS TSO commands **PJREQADD** and **PJREQMOD**.

PPJ0026 - Invalid years displayed by PJS ISPF calendar dialogs.

PPJ0027 - Message **PJS254E** issued by PJS TSO command **PJEVRSET**.

PPJ0028 - System console messages have the wrong route and descriptor codes.

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.