CA Panvalet® TSO Option

User Guide

r14.6
This Documentation, which includes embedded help systems and electronically distributed materials, (hereinafter referred to as the “Documentation”) is for your informational purposes only and is subject to change or withdrawal by CA at any time.

This Documentation may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of CA. This Documentation is confidential and proprietary information of CA and may not be disclosed by you or used for any purpose other than as may be permitted in (i) a separate agreement between you and CA governing your use of the CA software to which the Documentation relates; or (ii) a separate confidentiality agreement between you and CA.

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

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

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

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

The manufacturer of this Documentation is CA.

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

Copyright © 2004 CA. All rights reserved. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.
Contact CA Technologies

Contact CA Support

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

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

Providing Feedback About Product Documentation

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

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

55

Appendix B: Exit Facilities

69

6 User Guide
How to Use PANEXIT .................................................................................................................. 75
  Input Mode-PAN ADD/STORE .................................................................................................. 76
  Exit Directory Entry (DIRENTRY) .......................................................................................... 77
  Linking TSO Exits .................................................................................................................... 77
Codes Passed Between PANTSOEX and PAN............................................................................. 78
  User Exit Notes ....................................................................................................................... 80
Modifying Installation Defaults ................................................................................................. 81
Suppressing Subcommands ........................................................................................................ 81
  Examples .................................................................................................................................. 81
Sequence Numbering and Retrieval ............................................................................................. 82
  Formatting Options for Retrieve or Display ............................................................................. 82
  Example ................................................................................................................................... 82
  Formatting Options for Add or Store ....................................................................................... 83
Defining Work Data Set Attributes ............................................................................................ 83
  Deleting Input Data Set, Updating Date of Last Access .......................................................... 83
  Setting Block Size for Retrieve Data Sets .............................................................................. 84
Setting the Number of Blocks to Be Allocated for Retrieve Data Sets ........................................ 84

Index ................................................................................................................................. 85
Chapter 1: Introduction

AllFusion CA-Panvalet Option for TSO lets you develop programs online. Under TSO, the command processor performs storage and retrieval functions of all programs and data stored in the AllFusion CA-Panvalet library.

AllFusion CA-Panvalet Option for TSO:
- Executes concurrently in multiple TSO regions and on shared DASD among multiple CPUs.
- Invokes commands entered at the user's terminal during a TSO logon session and uses TSO prompting techniques.
- Provides commands to add and maintain AllFusion CA-Panvalet for z/OS members, which can be retrieved into a TSO sequential data set or a PDS. Likewise, both TSO sequential data sets and PDS members can be added to an AllFusion CA-Panvalet for z/OS library.
- Provides the capability to change and display member attributes.
- Allows display of all or part of an AllFusion CA-Panvalet for z/OS member.
- Supports member LOCK and UNLOCK commands.
- Supports TSO submitted batch jobs referencing TSO formatted members.

This guide describes how to use the AllFusion CA-Panvalet Option for TSO. This chapter describes the files that the PAN Command Processor uses in processing AllFusion CA-Panvalet library members.

This section contains the following topics:

File Requirements and Options (see page 10)
Getting Started (see page 14)
File Requirements and Options

The PAN Command Processor uses the ddnames PANDD1, PANDD2, and PANDD3 for input and output library/data files. Messages are written to the user's terminal.
Library File (PANDD1)

The AllFusion CA-Panvalet library resides on a direct access storage device and contains source programs, object programs, job control, and card image data files. It is logically segmented into library members referred to by name in AllFusion CA-Panvalet commands.

The PAN Command Processor normally uses the ddname PANDD1 to specify the AllFusion CA-Panvalet library file. The library can reside on any z/OS supported DASD. PAN#4 allocates space and formats the blocks. PAN#4 is normally executed in a batch environment but can be executed under TSO using a CALL command. You can have multiple libraries, and each can reside on multiple DASD volumes. If you use multiple libraries, ensure that they have different data set names. If you want a multi-volume library, create it in a batch PAN#4 execution rather than invoking it from the user terminal.

PANDD1 should normally be a catalogued data set. Support is provided for shared DASD operation on all DASD devices. Always use the library data set with DISP=SHR.

See Execution Under TSO in the chapter "PAN Command Processor" for information about initializing the AllFusion CA-Panvalet library file by calling PAN#4 from the user terminal and using a batch PAN#4 execution.
Output Work File (PANDD2)

The PAN Command Processor uses the ddname PANDD2 for its output work file. Any data selected by a PAN RETRIEVE command is placed on this file. This work file is normally a catalogued TSO sequential data set with the following naming standard:

```
userid.pannme.type
```

Following is a description of these options:

- **userid**
  - The user's logon ID.

- **panname**
  - The AllFusion CA-Panvalet member you want to retrieve. If the member name is greater than 10 characters, enter in the first eight characters (high-order) of the member name you want to retrieve.

- **type**
  - The language type converted to a standard TSO type qualifier as follows:

<table>
<thead>
<tr>
<th>Format</th>
<th>TSO Qualifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALC, BAL</td>
<td>ASM</td>
</tr>
<tr>
<td>COBOL, COBOL-72, ANSCOBOL</td>
<td>COBOL</td>
</tr>
<tr>
<td>FORTRAN</td>
<td>FORT</td>
</tr>
<tr>
<td>PL/I, PL/1</td>
<td>PLI</td>
</tr>
<tr>
<td>OBJECT</td>
<td>OBJ</td>
</tr>
<tr>
<td>JCL</td>
<td>CNTL</td>
</tr>
<tr>
<td>AUTOCODER, RPG, DATA, and OTHER</td>
<td>DATA</td>
</tr>
<tr>
<td>USER1, USER2, USER3, USER4, USER180, USER780</td>
<td>Set to the value specified in PVOPT</td>
</tr>
</tbody>
</table>

Use the DATASET operand on the RETRIEVE request to modify this naming convention.

- If the PANDD2 data set is new, the installation-option block size and record length of 80 are used.
- If the PANDD2 data set is old and you are reusing it, the original block size is used.
- If the PANDD2 data set refers to a member of a PDS, the PDS must be an existing PDS.

The PANDD2 data set is always written as a RECFM=FB, RECLN=80 file.
Input Work File (PANDD3)

The PAN Command Processor uses the ddname PANDD3 for the input work file. Any data selected by a PAN ADD or a PAN STORE command is read from this file. This work file is normally a cataloged TSO sequential data set with the following naming standard:

\[ \text{userid} \_ \text{panname} \_ \text{type} \]

Following is a description of these options:

- **userid**
  - The user's logon ID.

- **panname**
  - The AllFusion CA-Panvalet member you want to add or store (maximum of eight characters).

- **type**
  - The TSO type qualifier.

The PANDD3 data set must be old and cannot be empty. If several type qualifiers exist for the desired data set name, you are prompted to choose one. Use the DATASET operand with the ADD or STORE request to modify the naming convention.

The PANDD3 file is normally expected to be RECFM=FB, RECLN=80 file. If a VB file is encountered, the first 4 bytes of each record are dropped, and a maximum of 80 bytes are accepted as input.

User's Terminal

Informational, error, and termination messages from the PAN Command Processor are listed directly to the user's terminal. Input resulting from prompt messages is accepted directly from the user's terminal.
Getting Started

The rest of this guide is divided into several chapters to provide information about:

- Subcommands of the PAN Command Processor
- How to interface between TSO and Batch processing
- How to execute AllFusion CA-Panvalet under TSO
- The syntax of the PAN subcommands and a keyword summary for the subcommands
- Sample TSO sessions
- Exit facilities
Chapter 2: PAN Command Processor

The PAN Command Processor is structured to serve both programmers and programming management in processing AllFusion CA-Panvalet library members.

An exit facility is provided to accommodate user-written routines in the PAN Command Processor. See the appendix "Exit Facilities" for more information about exits.

This section contains the following topics:

- **PAN Subcommands** (see page 15)
- **Interface between TSO/Batch Processing** (see page 23)
- **Execution under TSO** (see page 26)

**PAN Subcommands**

A description of the subcommands of the PAN Command Processor follows:

**ADD**

The ADD subcommand adds TSO sequential data sets or PDS members to the AllFusion CA-Panvalet library. Required information includes the AllFusion CA-Panvalet member name and the language format. Data sets added to the library with the formats ALC, COBOL, PL/1, FORTRAN, JCL, or any ZTYPEs that mimic the mentioned languages are normally stored with AllFusion CA-Panvalet internal format, retaining the sequence NUMBER field. Other language types are stored with AllFusion CA-Panvalet internal format.

**Syntax**

```
PAN ADD NAME {FORMAT( )}
[DATASET( )]
[DDNAME( )]
[USER( )]
[NOFORMAT]
[PANVALET]
[TSO ]
[DELETE]
[KEEP ]
[CONTROL( )]
```
Parameters

PAN ADD

The minimum abbreviation is PAN A.

Name

The member name of the AllFusion CA-Panvalet member name you want to add.

When you initially enter a member into the library, assign the member a unique name. This name is used to reference the member for retrievals. The name can consist of ten alphanumeric or special characters in any order or combination (A-Z, 0-9, #, $, or @).

See Keyword Summary for additional information.

FORMAT( )

Required. Specifies the library formats. The AllFusion CA-Panvalet name that you add with certain TSO formats is stored according to one of the following AllFusion CA-Panvalet formats:

<table>
<thead>
<tr>
<th>TSO Format</th>
<th>AllFusion CA-Panvalet Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLI, PLIF, IPLI, PL/I, or PL/1</td>
<td>PL/I</td>
</tr>
<tr>
<td>CNTL</td>
<td>JCL</td>
</tr>
<tr>
<td>TEXT, BASIC, CLIST or GOFORT</td>
<td>OTHER</td>
</tr>
</tbody>
</table>
For a complete list of available formats, see Keyword Summary (default naming conventions) later in this chapter.

The minimum abbreviation is F.

**DATASET( )**

Optional. Specifies the input data set name. The minimum abbreviation is D.

**DDNAME( )**

Optional. Specifies an alternate ddname for the AllFusion CA-Panvalet library. The minimum abbreviation is DD.

**USER( )**

Optional. Specifies a user code for the library member. The minimum abbreviation is U.

**NOFORMAT**

Optional. Requests that neither special TSO handling nor AllFusion CA-Panvalet handling is performed, but that each entire statement is stored. The minimum abbreviation is N.

**PANVALET**

Optional. Requests that special TSO handling is not performed, but that each statement is stored according to the AllFusion CA-Panvalet internal format. The minimum abbreviation is P.

**TSO**

Optional. Requests that each statement is stored with TSO internal format. The minimum abbreviation is T.

**DELETE**

Optional. Requests that the input data set is deleted after a successful add. The minimum abbreviation is DE.

**KEEP**

Optional. Requests that the input data set is kept after a successful add. The minimum abbreviation is K.

**CONTROL( )**

Optional. Specifies a library control value needed to successfully perform the add if the subcommand is suppressed. The minimum abbreviation is CON.

* The installation default for internal formatting can be AllFusion CA-Panvalet or TSO. You can use the opposite parameter to override the default at execution time.

** The installation default for disposition of the input data set can be KEEP or DELETE. You can use the opposite parameter to override the default at execution time.
### Examples

This section provides examples of how to code the ADD command:

- Specify the following to add a TSO data set with a data set name of userid.PAYROLL5.ASM to the AllFusion CA-Panvalet library under the name of PAYROLL5, using BAL TSO internal format. The input data set is deleted if the add is successful.

  ```
  PAN A PAYROLL5 F(BAL) TSO DE
  ```

- If TSO is the installation default parameter, specify the following to add a TSO data set with a data set name of userid.V2DATERPT.COBOL to the AllFusion CA-Panvalet library under the name of V3DATERPT. Uses a COBOL TSO internal format and assigns a user code of 32 to the new member.

  ```
  PAN A V3DATERPT F(COBOL)
  DA(V2DATERPT) U(32)
  ```

- Specify the following to add a TSO data set with a data set name of SFTWRE.JCLTEST.CNTL to the AllFusion CA-Panvalet library under the name of TSTNEWJCL, using a JCL AllFusion CA-Panvalet internal format.

  ```
  PAN AD TSTNEWJCL
  DA('SFTWRE.JCLTEST.CNTL') F(JCL) PAN
  ```

### CHANGE

The CHANGE subcommand can apply, alter, or remove several AllFusion CA-Panvalet member attributes. These are changes in status, user code, level number, or user comment. For further information on the purpose of these subcommands, see the User Guide.

### Syntax

```
PAN CHANGE           NAME
{STATUS(  )  }
{USER(  )    }
{LEVEL(  )   }
{COMMENT(' ')}
[DDNAME(  )]
[ACCESS(  )]
[CONTROL(  )]
```
Parameters

PAN CHANGE
The minimum abbreviation is PAN CH.

Name
The member name is the AllFusion CA-Panvalet member name selected for the change.

See the Keyword Summary section for additional information.

STATUS( )
Required. Specifies a status change. The minimum abbreviation is S.

USER( )
Required. Specifies a user code or user security level change. The minimum abbreviation is U.

LEVEL( )
Required. Specifies a level change. The minimum abbreviation is L.

COMMENT( ' ' )
Required. Specifies a user comment. The minimum abbreviation is COM.

DDNAME( )
Optional. Specifies an alternate ddname for the AllFusion CA-Panvalet library. The minimum abbreviation is DD.

ACCESS( )
Optional. Specifies a security value to allow access to a user security-protected AllFusion CA-Panvalet library member. The minimum abbreviation is A.

CONTROL( )
Optional. Specifies a library control value needed to successfully perform the change if the subcommand is suppressed. The minimum abbreviation is CON.
Examples

This section provides examples of how to code the CHANGE subcommand:

- Specify the following to perform a status change to production on DATAREC1 on the AllFusion CA-Panvalet library.
  
  PAN CHA DATAREC1 S(Prod)

- Specify the following to perform a level change in XYZFILE from 65 to 1, provided 65 matches the current level number of the member. The character A indicates that the member is user security protected. The PAN Command Processor prompts the user for this value on the print bypass mode.
  
  PAN CHANGE XYZFILE L(65 1) A

- Specify the following to change the user code of QUERYSET10 on the AllFusion CA-Panvalet library to a value of 6.
  
  PAN CH QUERYSET10 U(6)

- Specify the following to attach the user comment specified in the quotes to PAYRLCALV9.
  
  PAN CH PAYRLCALV9
  COM('CALIF PAYROLL RUN VERSION 9
  09-15-73')

COPY

The COPY subcommand creates a duplicate member on the AllFusion CA-Panvalet library. You give the copy a new name and AllFusion CA-Panvalet places the copy in test, enabled status. Required information includes the original AllFusion CA-Panvalet member name and the new member name you want to create.

DISPLAY

The DISPLAY subcommand displays an AllFusion CA-Panvalet library member at the terminal. You can display the requested member in its entirety or in various selected portions.

LISTDS

The LISTDS subcommand lets you inspect the attributes of a CA-Panvalet library member. The CA-Panvalet member name is required. Information for the requested member is listed to the user’s terminal and includes level number, user code, status, record length, language, blocks used, number of statements, dates of last maintenance and last access, last action taken, and percent utilization.
LOCK

The LOCK subcommand lets you lock a member to your TSO user ID. Once a member is locked, only the user it is locked to can alter it. Other users cannot modify the member’s data, attributes, or comment in any way (regardless of status). Locking a member is treated as an action against that member. For locked members, AllFusion CA-Panvalet reports the last action value as LOC. Locking a member does not modify any of the member's current attributes. For more information about the LOCK subcommand, see the "Executing PAN#1" chapter in the User Guide.

PRINT

The PRINT subcommand prints a selective directory list in abbreviated format (72 characters per line) at the user’s terminal. You can select on namekey, status, user code, or language type. Regardless of the selection, the listing appears in name sequence. Other TSO or batch users can access the AllFusion CA-Panvalet library while this subcommand is being performed. Subtotals or summaries are not produced. You must supply the proper control code, as in PAN#2, for this subcommand to execute successfully.

RENAME

The RENAME subcommand changes the name of an AllFusion CA-Panvalet library member. Required information includes the old member name and the new member name.

RETRIEVE

The RETRIEVE subcommand copies a CA-Panvalet library member from the library to a TSO sequential data set or to a PDS member. A standard TSO dsname is normally generated based on the user ID, the AllFusion CA-Panvalet member name, and the AllFusion CA-Panvalet language format. The TSO EDIT or the interactive compilers can read the retrieved data.

STORE

The STORE subcommand updates an AllFusion CA-Panvalet library member by replacing a TSO sequential data set or PDS member. Required information includes the AllFusion CA-Panvalet member name and the current level number.
UNLOCK

The UNLOCK subcommand removes the LOCK indicator from an AllFusion CA-Panvalet member. See LOCK, earlier in this chapter.
Interface between TSO/Batch Processing

A number of considerations have been made for the TSO user who wants to reference TSO formatted members on the AllFusion CA-Panvalet library in a batch mode (submitting a job which executes PAN#1).

The special formatting option is part of the ++ADD command. The ++ADD command in the PAN-TSO environment supports the optional parameter (TSO), which you can supply in place of the NOFORMAT parameter. This parameter requests special TSO internal storage and processing for the member you are adding to the AllFusion CA-Panvalet library. This operand is ignored for formats other than:

- ANSCOBOL
- BAL (or ALC)
- COBOL
- FORTRAN
- JCL
- PL/I (or PL/1)

The special processing features for a TSO library member with PAN#1 follow:

- The program, when added to the library in a batch PAN#1 run, is resequenced by tens (10s), and these sequence numbers are stored on the library.
- When you retrieve the member using batch PAN#1 or the TSO PAN Command Processor, a special comment record is developed to inform the user of the AllFusion CA-Panvalet name, the LEVEL number of the member, and the Date of Last Maintenance. This comment is a separate statement, and its sequence number is all zeros.
- For the COBOL format with the TSO option, the ID portion of the statement is blank after the first three cards.
- You can make a batch mode update of a program stored with TSO format using PAN#1. The ++C updating technique is used. For example, the following deletes all statements starting with 4090 and including 5032:
  
  ++C 4090,5032

  This next example inserts, after the statement number 6050, the cards:

  ++C 6050
  ...statements to insert

  And this last example deletes all the statements starting with 7032 and including 7040, and inserts the cards which follow in the place of the deleted statements:

  ++C 7032,7040
  ...statements to replace
The updated program is renumbered by tens. A compile of the updated member shows the new sequence numbers.
Special Processing Features

The special processing features for a TSO library member with the PAN Command Processor are described in the following topics. The PAN Command Processor can retrieve or display the member in a variety of ways as follows:

- **A PAN R or PAN D with PANVALET,NOEXPAND parameters or defaults performs the retrieval or display exactly as a PAN#1 ++WRITE PRINT.**

- **A PAN R or PAN D with PANVALET,EXPAND parameters or defaults performs the retrieval or display exactly as a PAN#1 ++WRITE WORK (except for PL/1). Multi-level ++INCLUDEs, if present, are expanded.**

- **A PAN R or PAN D with TSO,NOEXPAND parameters or defaults retrieves the following format types with a generated sequence field, sequenced by tens, filling the entire sequence field: ASM, COBOL, FORT, PLI, CNTL. Members that are already stored with TSO format use the stored sequence numbers. Other formats or any language types stored as NOFORM are retrieved exactly as a PAN#1 ++WRITE PRINT.**

- **A PAN R or PAN D with TSO,EXPAND parameters or defaults retrieves the following format types with a generated sequence field, sequenced by tens, filling the entire sequenced field: ASM, COBOL, FORT, PLI, CNTL, or any ZTYPEs that mimic the mentioned languages. Embedded ++INCLUDEs are expanded and sequenced inline with the original member. Members already stored with TSO format are resequenced by tens from the beginning rather than using the stored sequence numbers. (This is done to accommodate the inline sequencing of the expanded includes.) Other formats or any language type stored as NOFORMAT is retrieved with AllFusion CA-Panvalet sequencing, if applicable. Multi-level ++INCLUDEs are expanded and sequenced, if applicable, inline with the original member. The expansion of the ++INCLUDEs is performed as follows. The first level include (or outermost include) is expanded with a special header and trailer comment to mark the beginning and end of the included member. Additional levels of includes are expanded normally.**

The format of the header comment follows:

<table>
<thead>
<tr>
<th>type</th>
<th>column</th>
</tr>
</thead>
<tbody>
<tr>
<td>COBOL, RPG</td>
<td>1 2 3 7 8 7172</td>
</tr>
<tr>
<td>FORT</td>
<td>C 1 INCLUDE++</td>
</tr>
<tr>
<td>PLI</td>
<td>/ * INCLUDE++</td>
</tr>
<tr>
<td>CNTL</td>
<td>/ * INCLUDE++</td>
</tr>
<tr>
<td>ASM and all others</td>
<td>* INCLUDE++</td>
</tr>
</tbody>
</table>

The format of the trailer comment is the same except for the key phrase END INCLUDE++, which starts in column 8 of the trailer record.
Important! You must not alter the comments under TSO EDIT if the member is to be properly stored back on the AllFusion CA-Panvalet library.

The resulting output member is compatible, if TSO sequencing is applicable, with the TSO interactive compilers or with TSO EDIT.

The following processing techniques are used when adding or storing a member with the PAN Command Processor:

- A PAN A or PAN S with the PANVALET operand or default performs normal AllFusion CA-Panvalet formatting (sequence and identification fields are dropped when possible). Any expanded includes, marked by the header and trailer comments generated by a PAN R, are compressed back to a single ++INCLUDE statement as the member is written to the AllFusion CA-Panvalet library.

- A PAN A or PAN S with the TSO operand or default retains existing sequence numbers for ALC, ANSCOBOL, COBOL, FORTRAN, PL/I, or JCL library formats (unless currently stored as NOFORMAT). The specially expanded includes are compressed back to single ++INCLUDE statements.

Note: TSO sequenced members should not be larger than 99,999 physical statements; therefore, retrieves or displays should not go beyond physical statement 99,999. Using AllFusion CA-Panvalet sequencing allows the retrieval or display beyond physical statement 99,999.

Execution under TSO

This section describes execution under TSO.
PAN Command Processor

Execute the PAN Command Processor by invoking the command (PAN) and the desired subcommand with its operands directly from the user’s terminal during a TSO logon session. To execute in this fashion, the PAN load module should be located in either the TSO command library or in a private command library. To concatenate this library to the TSO command library, use a //STEPLIB DD statement in the user’s logon procedure.

To use the PAN Command Processor during a terminal session, you must first allocate the ddname of PANDD1. You can accomplish this in either of two ways:

- Modify the user’s logon procedure to include the allocation using a DD statement.
- Issue the following allocation command during the terminal session after the logon, where x is the DSNAME for the AllFusion CA-Panvalet library:

  `ALLOCATE FILE(PANDD1) DATASET(x) SHR`

We recommend the first method because it is automatic with every logon and transparent to the user.

For easy access to multiple AllFusion CA-Panvalet libraries during a single logon session, assign additional ddnames to the libraries during or after logon and use the ddname operand with each PAN request to indicate the ddname of the library you want to access. You can use any appropriate ddname, except PANDD2 and PANDD3.

Syntax Format

This topic describes the PAN Command Processor syntax format.

<table>
<thead>
<tr>
<th>Pan Subcommand</th>
<th>Name</th>
<th>Keywords/Values</th>
</tr>
</thead>
</table>

PAN Subcommand

The command is PAN, followed by a space and then the subcommand. The minimum abbreviation for a subcommand is variable, and it is followed by a space.

Name

A one- to ten-character AllFusion CA-Panvalet member name that the subcommand processes. It is followed by a space.
Keywords/Values

Keywords are non-positional and delimited by a space or spaces from each other and from their values (if any). You can abbreviate each keyword as described in the keyword summary. You can specify keywords in any order.

Keyword values assign meaning or quantity that directs or qualifies the action of the keyword; an operand. The value of a keyword must follow right after the keyword.

Required keywords are enclosed in braces:

(required)

Mutually exclusive required keywords are stacked in braces:

{ one of these }
{ keywords     }
{ is required  }

Optional keywords are enclosed in brackets:

[Optional]

Mutually exclusive, optional keywords are stacked in brackets:

[ one of these ]
[ keywords     ]
[ can be used  ]

Variables (for example, an AllFusion CA-Panvalet member name or keyword value) are shown in lower case.
Batch Program

You can execute the PAN Command Processor only while TSO is active; the batch AllFusion CA-Panvalet programs (PAN#1, PAN#2, PAN#3, PAN#4, PAN#6, PAN#7, PAN#8) from the user's terminal should generally be avoided unless being used with the SUBMIT feature. This is recommended for reasons of efficiency only. If you must execute the batch program under TSO (using the CALL command), route the SYSPRINT data set to a disk file. You can list this disk file after the program is finished rather than routing SYSPRINT directly to the user's terminal.

To execute a PAN#4 under TSO for the purpose of creating a small testing AllFusion CA-Panvalet library, issue the following series of TSO commands either one by one or by defining the commands in a command procedure:

ALLOC FI(SYSPRINT) DA(PAN4.LIST) NEW SP(105) BL(1210)
ALLOC FI(SYSIN) DA(*)
ALLOC FI(PAN0D1) DA(PANVALET.LIBRARY) NEW SP(76) BL(3156)
CALL 'SYS1.LINKLIB(PAN4)'
++CLEAR
++SUPPRESS ++RENAME
LIST PAN4.LIST 2 12 NONUM COL(1:90)
DELETE (PAN4.LIST)
Internal Organization

The PAN Command Processor performs its interactive functions using the standard IBM service routines, IKJSCAN and IKJPARS. Data set allocations (other than the AllFusion CA-Panvalet library allocation) are performed dynamically using IKJDFLT and IKJDAIR.

All interactive processing and allocations are performed before any attempt is made to access the AllFusion CA-Panvalet library. This ensures that the library is updated or accessed by a single user for the shortest possible time.

When the library is accessed, a PAN attention interrupt routine is in effect. This exit routine is incorporated to attempt to close any opened files in the event of an attention interrupt. The exit is not in effect while PAN is processing interactive data or while performing dynamic allocation. The first attention interrupt causes the message PAN INTERRUPTED to print at the terminal. You can take one of three courses of action:

- Enter a null line. This causes the AllFusion CA-Panvalet Option for TSO to continue processing as normal. Note that the contents of any of the buffers can be lost.
- Give a second attention interrupt. The AllFusion CA-Panvalet Option for TSO leaves the user in the READY mode.
- Enter any non-blank characters. This is necessary to clear a footprint, if the FTP option is in use, before going to the READY mode.

Messages are normally written to the user’s terminal after library processing is complete.
Chapter 3: PAN Subcommands

This chapter describes all the PAN Command Processor subcommands and keywords. For more information about subcommands, see the chapter "PAN Command Processor."

This section contains the following topics:

ADD (see page 31)
CHANGE (see page 31)
PAN COPY (see page 31)
PAN DISPLAY (see page 32)
PAN LISTDS (see page 35)
PAN LOCK (see page 36)
PAN PRINT (see page 37)
PAN RENAME (see page 38)
PAN RETRIEVE (see page 38)
PAN STORE (see page 40)
PAN UNLOCK (see page 42)
Keyword Summary (see page 43)

ADD

PAN ADD requests to add a member to the AllFusion CA-Panvalet library.

CHANGE

PAN CHANGE requests to perform a level, status, user code, or user comment change on an AllFusion CA-Panvalet library member.

PAN COPY

PAN COPY is a request to copy an AllFusion CA-Panvalet library member.

Syntax

\[
\text{PAN COPY} \quad \text{NAME1} \quad \text{NAME2} \quad \text{[DDNAME( )]} \\
\phantom{\text{PAN COPY} \quad \text{NAME1} \quad \text{NAME2} \quad \text{[DDNAME( )]}} \quad \text{[ACCESS( )]} \\
\phantom{\text{PAN COPY} \quad \text{NAME1} \quad \text{NAME2} \quad \text{[DDNAME( )]} \quad \text{[ACCESS( )]}} \quad \text{[CONTROL( )]}
\]
Parameters

PAN COPY

The minimum abbreviation is PAN C.

Names

NAME1 is the original AllFusion CA-Panvalet member name from which you want to copy. NAME2 is the new AllFusion CA-Panvalet member name to which you want to copy.

See Keyword Summary for additional information.

DDNAME( )

Optional. Specifies an alternate ddname for the AllFusion CA-Panvalet library. The minimum abbreviation is DD.

ACCESS( )

Optional. Specifies a security value to allow access to a user security protected AllFusion CA-Panvalet library member. The minimum abbreviation is A.

CONTROL( )

Optional. Specifies a library control value needed to successfully perform the copy if the subcommand is suppressed. The minimum abbreviation is CON.

Examples

This section provides examples of how to code the COPY subcommand:

- Specify the following to create a copy of CREDRPT06 on the AllFusion CA-Panvalet library. The new copy is called CREDRPT07 and is in test, enabled status.

  PAN C CREDRPT06 CREDRPT07

- Specify the following to copy INVFILEDER to TESTINVFIL, placing TESTINVFIL in test, enabled status. INVFILEDER is a user security protected data set which requires an ACCESS code to successfully reference the member. The user is prompted for the value in print bypass mode.

  PAN C INVFILEDER TESTINVFIL

PAN DISPLAY

PAN DISPLAY is a request to display an AllFusion CA-Panvalet library member.
Syntax

PAN DISPLAY  NAME  [DDNAME( )]
[EXPAND   ]
[NOEXPAND]
[PANVALET]
[TSO      ]
[ACCESS( )]
[CONTROL( )]
[SEQ1( )]
[SEQ2( )]
Parameters

PAN DISPLAY
The minimum abbreviation is PAN D.

Name
The member name is the AllFusion CA-Panvalet member name you want to display.

See Keyword Summary for additional information.

DDNAME
Optional. Specifies an alternate ddname for the AllFusion CA-Panvalet library. The minimum abbreviation is DD.

EXPAND*
Optional. Requests the expansion of embedded AllFusion CA-Panvalet ++INCLUDE statements during retrieval from the AllFusion CA-Panvalet library for display. The minimum abbreviation is E.

NOEXPAND*
Optional. Requests to leave embedded AllFusion CA-Panvalet ++INCLUDE statements unexpanded during retrieval from the AllFusion CA-Panvalet library for display. The minimum abbreviation is N.

PANVALET**
Optional. Requests normal AllFusion CA-Panvalet sequencing. The minimum abbreviation is P.

TSO**
Optional. Request TSO sequencing of a member. The minimum abbreviation is T.

ACCESS( )
Optional. Specifies a security value to allow access to a user security protected AllFusion CA-Panvalet library member. The minimum abbreviation is A.

CONTROL( )
Optional. Specifies a library control value needed to successfully perform the display if the subcommand is suppressed. The minimum abbreviation is CON.

SEQ1( )
Optional. Requests the display begin at a specific statement.

SEQ2( )
Optional. Requests the display end after this statement.

Note: Specifying only SEQ1 causes the display to start with the statement indicated and continue until interrupted, or until reaching the end of the data set, or until the 99,999th statement is displayed using TSO sequencing because the sequencing is by tens.
* The installation default for display of ++INCLUDEs can be either EXPAND or NOEXPAND. You can use the opposite parameter to override the default at execution time.

** The installation default for internal formatting can be PANVALET or TSO. You can use the opposite parameter to override the default at execution time.

**PAN LISTDS**

PAN LISTDS is a request to list the attributes of an AllFusion CA-Panvalet library member.

**Syntax**

```
PAN LISTDS           NAME               [COMMENT]
LD [DDNAME(  )]
[ACCESS(  )]
[CONTROL(  )]
```

**Parameters**

PAN LISTDS

The minimum abbreviation is PAN L.

Name

The member name is the AllFusion CA-Panvalet member name you want to list.

See Keyword Summary for additional information.

COMMENT

Optional. Requests to have the library member's user comment listed to the terminal. The minimum abbreviation is COM.

DDNAME( )

Optional. Specifies an alternate ddname for the AllFusion CA-Panvalet library. The minimum abbreviation is DDM.

ACCESS( )

Optional. Specifies a security value to allow access to a user security protected AllFusion CA-Panvalet library member. The minimum abbreviation is A.

CONTROL( )

Optional. Specifies a library control value needed to successfully perform the list if the subcommand is suppressed. The minimum abbreviation is CON.
Example

Specify the following to list one heading line to the user's terminal, one line of directory information for AllFusion CA-Panvalet member TSOPANVAL, and one line of user comment, if present:

PAN LD TSOPANVAL COM

PAN LOCK

The PAN LOCK subcommand locks an AllFusion CA-Panvalet member to your TSO user ID. Once a member is locked, only the user it is locked to can alter it. Other users cannot modify the member's data, attributes, or comment in any way (regardless of status).

See the User Guide for more information about the LOCK subcommand.

Syntax

PAN LOCK              NAME1              [DDNAME(   )]
                     [ACCESS(   )]
                     [CONTROL(   )]

Parameters

PAN LOCK

The minimum abbreviation for this subcommand is PAN LO.

Name

NAME1 specifies the member name you want to lock.

See Keyword Summary for additional information.

DDNAME( )

Optional. Specifies an alternate ddname for the AllFusion CA-Panvalet library. The minimum abbreviation is DD.

ACCESS( )

Optional. Specifies a security value to allow access to a user security protected AllFusion CA-Panvalet library member. The minimum abbreviation is A.

CONTROL( )

Optional. Specifies a library control value needed to successfully perform the lock for a suppressed subcommand. The minimum abbreviation is CON.
**PAN PRINT**

PAN PRINT is a request to print a selective directory list at the user terminal.

**Syntax**

```
PAN PRINT                     {CONTROL(  )}
{NAME(  )}
{STATUS( )}
{USER( )}
{TYPE( )}
[COMMENT]
[DDNAME( )]
```

**Parameters**

PAN PRINT

The minimum abbreviation is PAN P.

See Keyword Summary for additional information.

**CONTROL( )**

Required. Specifies a library control code. The minimum abbreviation is CON.

Note: If the control code is zero, specify CONTROL(0).

**NAME( )**

Required. Specifies a selection on namekey. The minimum abbreviation is N.

**STATUS( )**

Required. Specifies a selection on status. The minimum abbreviation is S.

**USER( )**

Required. Specifies a selection on user code or range of user codes. The minimum abbreviation is U.

**TYPE( )**

Required. Specifies a selection on language type. The minimum abbreviation is T.

**COMMENT**

Optional. Specifies that user comments are also listed, when present. The minimum abbreviation is COM.

**DDNAME( )**

Optional. Specifies an alternate ddname for the CA-Panvalet library. The minimum abbreviation is DD.
PAN RENAME

PAN RENAME is a request to rename an AllFusion CA-Panvalet library member.

Syntax

```
PAN RENAME   NAME1   NAME2        [DDNAME( )]
               [ACCESS( )]
               [CONTROL( )]
```

Examples

This section provides examples of how to code the RENAME subcommand:

- Specify the following to rename AllFusion CA-Panvalet library member KJP5009T to KJP5009V8.
  
PAN REN KJP5009T KJP5009V8

- Specify the following to rename AllFusion CA-Panvalet library member XYZW3 to XYZW5. If member XYZW3 is secured, the user is prompted for an access code.
  
PAN RENAME XYZW3 XYZW5 A

PAN RETRIEVE

PAN RETRIEVE is a request to retrieve an AllFusion CA-Panvalet library member.

Syntax

```
PAN RETRIEVE   NAME               [DATASET( )]
               [DDNAME( )]
               [EXPAND   ]
               [NOEXPAND]
               [PANVALET]
               [TSO      ]
               [VOLUME( )]
               [ACCESS( )]
               [CONTROL( )]
```
Parameters

PAN RETRIEVE

The minimum abbreviation is PAN R.

Name

The member name is the AllFusion CA-Panvalet member name you want to retrieve.

See Keyword Summary for additional information.

DATASET( )

Optional. Specifies the output data set name. The minimum abbreviation is DA.

DDNAME( )

Optional. Specifies an alternate ddname for the AllFusion CA-Panvalet library. The minimum abbreviation is DD.

EXPAND*

Optional. Requests expansion of embedded AllFusion CA-Panvalet ++INCLUDE statements during retrieval. The minimum abbreviation is E.

NOEXPAND*

Optional. Requests to leave embedded AllFusion CA-Panvalet ++INCLUDE statements unexpanded during retrieval. The minimum abbreviation is N.

PANVALET**

Optional. Requests normal AllFusion CA-Panvalet sequencing. The minimum abbreviation is P.

TSO**

Optional. Requests TSO sequencing of a member. The minimum abbreviation is T.

VOLUME( )

Optional. Requests to retrieve the member to a specific volume serial number. The minimum abbreviation is V.

ACCESS( )

Optional. Specifies a security value to access a user security protected AllFusion CA-Panvalet library member. The minimum abbreviation is A.

CONTROL( )

Optional. Specifies a library control value needed to successfully perform the retrieve if the subcommand is suppressed. The minimum abbreviation is CON.

* The installation default for retrieval of ++INCLUDEs can be EXPAND or NOEXPAND. You can use the opposite parameter to override the default at execution time.
** The installation default for internal formatting can be PANVALET or TSO. You can use the opposite parameter to override the default at execution time.

If an ICTL assembler statement is the first statement in a BAL format data set, then the AllFusion CA-Panvalet special comment record is not generated.

**Note:** When retrieving to a PDS, a RESERVE and a STEP ENQ are issued against the PDS for the duration of the retrieve function in progress.

**Note:** If you use a PROCESS compiler control statement as the first statement in a PL/1 format, the AllFusion CA-Panvalet special comment record is not generated.

**PAN STORE**

PAN STORE is a request to update an AllFusion CA-Panvalet library member by replacement.

**Syntax**

```
PAN STORE          NAME               {LEVEL(  )}
                   [RECLN(  )]
                   [DATASET(  )]
                   [DDNAME(  )]
                   [PANVALET(  )]
                   [TSO         ]
                   [DELETE]
                   [KEEP  ]
                   [ACCESS(  )]
                   [CONTROL(  )]
```
Parameters

PAN STORE
The minimum abbreviation is PAN S.

Name
The member name is the AllFusion CA-Panvalet member name you want to update.

See Keyword Summary for additional information.

LEVEL( )
Required. Specifies the current level number of the AllFusion CA-Panvalet library member you want to update. The minimum abbreviation is L.

RECLN( )
Optional. Request to change the logical record length of the existing member. The minimum abbreviation is R.

DATASET( )
Optional. Specifies the input data set name. The minimum abbreviation is DA.

DDNAME( )
Optional. Specifies an alternate ddname for the AllFusion CA-Panvalet library. The minimum abbreviation is DD.

PANVALET*
Optional. Requests AllFusion CA-Panvalet internal format processing. The minimum abbreviation is P.

TSO*
Optional. Requests TSO internal format processing. The minimum abbreviation is T.

DELETE**
Optional. Requests that the input data set is deleted after a successful store. The minimum abbreviation is DE.

KEEP**
Optional. Requests that the input data set is kept after a successful store. The minimum abbreviation is K.

ACCESS( )
Optional. Specifies a security value to allow access to a user security protected AllFusion CA-Panvalet library member. The minimum abbreviation is A.

CONTROL( )
Optional. Specifies a library control value needed to successfully perform the store if the subcommand is suppressed. The minimum abbreviation is CON.
* The installation default for internal formatting can be PANVALET or TSO. You can use the opposite parameter to override the default at execution time.

** The installation default for disposition of the input data set can be KEEP or DELETE. You can use the opposite parameter to override the default at execution time.

** PAN UNLOCK **

The PAN UNLOCK subcommand unlocks an AllFusion CA-Panvalet member. Unlocking a member removes the LOCK indicator from an AllFusion CA-Panvalet member.

See the User Guide for more information about the UNLOCK subcommand.

** Syntax **

PAN UNLOCK NAME1 [DDNAME()] [ACCESS()] [CONTROL()]

** Parameters **

PAN UNLOCK

The minimum abbreviation for the subcommand is PAN UN.

** Name **

NAME1 specifies the AllFusion CA-Panvalet member name you want to UNLOCK.

See the Keyword Summary section for additional information.

** DDNAME() **

Optional. Specifies an alternate ddname for the AllFusion CA-Panvalet library. The minimum abbreviation is DD.

** ACCESS() **

Optional. Specifies a security value to allow access to a user security protected AllFusion CA-Panvalet library member. The minimum abbreviation is A.

** CONTROL() **

Optional. Specifies a library control value needed to successfully perform the UNLOCK for a suppressed subcommand. The minimum abbreviation is CON.
### Example

Specify the following to remove the LOCK indicator from PAY57:

```
PAN UNLOCK PAY57
```

PAY57 retains all of the characteristics that existed before you locked it.

### Keyword Summary

The following table is a summary of possible keyword usage with the AllFusion CA-Panvalet TSO Option subcommands. The minimum abbreviations of the subcommands are shown in bold text.

**Note:** Keywords used with PAN subcommands must be enclosed as a group in a single set of parentheses.

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Where Used</th>
<th>Action</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACCESS</td>
<td>Optional with COPY, RETRIEVE, STORE, RENAME, CHANGE, LISTDS, and DISPLAY subcommands.</td>
<td>Allows access to a user security protected AllFusion CA-Panvalet library member.</td>
<td>Numeric value one to five digits. If entered without parentheses, the access code is prompted for in bypass mode.</td>
</tr>
<tr>
<td>COMMENT</td>
<td>Optional with the LISTDS, CHANGE, and PRINT subcommands.</td>
<td>CHANGE subcommand—Indicates a request to add, replace, or delete a user comment record on an AllFusion CA-Panvalet library member.</td>
<td>CHANGE subcommand—Up to fifty positions of information of any type, entered in quotes.</td>
</tr>
</tbody>
</table>

LISTDS, PRINT subcommands—Indicates a request to have the user comment record listed to the user's terminal along with the normal information.
<table>
<thead>
<tr>
<th><strong>Keyword</strong></th>
<th><strong>Where Used</strong></th>
<th><strong>Action</strong></th>
<th><strong>Format</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>CONTROL</td>
<td>Optional with ADD, COPY, CHANGE, RENAME, RETRIEVE, STORE, LISTDS, and DISPLAY subcommands. Required with the PRINT subcommand.</td>
<td>Allows the execution of a suppressed subcommand or of a subcommand that is normally protected.</td>
<td>Numeric value one to five digits. The minimum abbreviation is CON. If entered without parentheses, the control code is prompted for in bypass mode.</td>
</tr>
<tr>
<td>DATASET</td>
<td>Optional with the ADD, RETRIEVE, and STORE subcommands.</td>
<td>Allows input or output data set naming other than standard default.</td>
<td>See DATASET NOTE, which follows this table.</td>
</tr>
<tr>
<td>DDNAME</td>
<td>Optional on the ADD, COPY, CHANGE, RENAME, RETRIEVE, STORE, LISTDS, PRINT, and DISPLAY subcommands.</td>
<td>Lets you use a ddname other than PANDD1 for the AllFusion CA-Panvalet library. You can use any ddname except PANDD2 or PANDD3. The ddnames must be allocated to the respective libraries prior to invoking PAN. If omitted, PANDD1 is used.</td>
<td>From one- to eight-position alphanumeric ddname allocated to an AllFusion CA-Panvalet library.</td>
</tr>
<tr>
<td>Keyword</td>
<td>Where Used</td>
<td>Action</td>
<td>Format</td>
</tr>
<tr>
<td>---------</td>
<td>------------</td>
<td>--------</td>
<td>--------</td>
</tr>
<tr>
<td>DELETE</td>
<td>Optional with the ADD and STORE subcommands.</td>
<td>Allows an override of the system installation option to keep an input data set after the ADD or STORE subcommand is successfully completed. If the system installation option is to delete the input data set, then this operand has no effect. If the data set keyword gives a quoted DSN or points to a PDS member, the input data set is not deleted.</td>
<td></td>
</tr>
<tr>
<td>EXPAND</td>
<td>Optional on the RETRIEVE and DISPLAY subcommands.</td>
<td>The embedded source level command, <code>++INCLUDE</code>, allows standardization and storage efficiency in an AllFusion CA-Panvalet library. Use of this parameter allows an override of the system installation option to not EXPAND includes during retrieval. If the installation default is to EXPAND includes, then this operand has no effect.</td>
<td></td>
</tr>
<tr>
<td>Keyword</td>
<td>Where Used</td>
<td>Action</td>
<td>Format</td>
</tr>
<tr>
<td>---------</td>
<td>------------</td>
<td>--------</td>
<td>--------</td>
</tr>
</tbody>
</table>
| FORMAT  | Required with the ADD subcommand. | Indicates the CA-Panvalet format used. See the User Guide for a description of the formats. If the TSO operand/default is used, the following language types are stored with the TSO sequence numbers and are flagged as having TSO internal format: ASM, COBOL, ANSCOBOL, FORTRAN, PL/I, CNTL, or any ZTYPEs that mimic the mentioned languages. **Note:** This also applies to USER1, USER2, USER3, USER4, USER180, and USER780. The value is set to the value specified in PVOPT. | The available values are:  
- AUTOCODER  
- ALC, BAL  
- COBOL, ANSCOBOL  
- COBOL-72  
- DATA3  
- FORTRAN, FORTRAN  
- JCL  
- OBJECT  
- OTHER  
- PL/I, PL/I  
- RPG  
- USER1, USER2, USER3, USER4, USER180, and USER780 (see Note under Action column)  
**Note:** The AllFusion CA-Panvalet TSO Option supports user formats and ZTYPEs. See the AllFusion CA-Panvalet for z/OS Getting Started. |
<p>| KEEP    | Optional with the ADD and STORE subcommands. | Allows an override of the system installation option to delete an input data set after the ADD or STORE subcommand is successfully completed. If the system installation option is to KEEP the input data set, then this operand has no effect. |</p>
<table>
<thead>
<tr>
<th>Keyword</th>
<th>Where Used</th>
<th>Action</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>LEVEL</strong></td>
<td>Required with the STORE subcommand, optional with the CHANGE subcommand.</td>
<td>STORE subcommand—Allows the input data set to replace an AllFusion CA-Panvalet library member in an UPDATE...ALL fashion. The value used is the value reported during the most recent RETRIEVE operation on that library member.</td>
<td>Numeric value one to three positions. You must supply one value (current-level) with the STORE request. You must supply two values (old-level, new-level) with the CHANGE request.</td>
</tr>
<tr>
<td><strong>CHANGE</strong></td>
<td></td>
<td>CHANGE subcommand—Allows a change in level number. The first value must match the current level before the requested level change can be made.</td>
<td></td>
</tr>
<tr>
<td><strong>NAME</strong></td>
<td>Optional with the PRINT subcommand.</td>
<td>Causes all AllFusion CA-Panvalet library members that begin with the named characters to be selected for the listing.</td>
<td>From one to ten alphanumeric and special characters.</td>
</tr>
<tr>
<td><strong>NOEXPAND</strong></td>
<td>Optional with the RETRIEVE and DISPLAY subcommands.</td>
<td>The embedded source level command, ++INCLUDE, allows standardization and storage efficiency in an AllFusion CA-Panvalet library. Use of this parameter allows an override of the system installation option to EXPAND INCLUDEs during retrieval. If the installation default is to not EXPAND INCLUDEs, then this operand has no effect.</td>
<td></td>
</tr>
<tr>
<td><strong>NOFORMAT</strong></td>
<td>Optional with the ADD subcommand.</td>
<td>Causes the library member to be identified by the requested format, yet the entire statement is stored with no alteration or stripping of fields. Normally this is not used.</td>
<td></td>
</tr>
<tr>
<td>Keyword</td>
<td>Where Used</td>
<td>Action</td>
<td>Format</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>PANVALET</td>
<td>Optional with the ADD, RETRIEVE, DISPLAY, and STORE subcommands.</td>
<td>ADD, STORE subcommands—Causes the library member to be added or stored with AllFusion CA-Panvalet internal format. Sequence and identification fields are stripped before storage on the library. Use of this parameter allows an override of the system installation option to use PANVALET internal format when applicable.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>RETRIEVE and DISPLAY subcommands: Causes the library member to be retrieved with normal AllFusion CA-Panvalet formatting. A five-position sequence field with sequencing by ones is used when applicable. Use of this parameter allows an override of the system installation option to retrieve members with TSO sequencing when possible. With this parameter specified, you can retrieve a maximum of 99,999 records from the library member.</td>
<td></td>
</tr>
<tr>
<td>Keyword</td>
<td>Where Used</td>
<td>Action</td>
<td>Format</td>
</tr>
<tr>
<td>---------</td>
<td>------------</td>
<td>--------</td>
<td>--------</td>
</tr>
<tr>
<td><strong>RECLN</strong></td>
<td>Optional with the STORE subcommands.</td>
<td>Causes the existing member’s record length to be changed. Use of this parameter allows an override of the existing member’s record length, and replaces it with the new record length. If the member’s record length is greater than 80 bytes, the language type must be DATA. The input member record length must match the RECLN specified in the PAN STORE RECLN(nnnn) command.</td>
<td>The record length is a one to four digit numeric value.</td>
</tr>
<tr>
<td><strong>SEQ 1, SEQ 2</strong></td>
<td>Optional with the DISPLAY subcommand.</td>
<td>Causes specific statements to be displayed on the terminal. SEQ1 indicates the first statement you want to display. SEQ2 indicates the last statement you want to display.</td>
<td></td>
</tr>
</tbody>
</table>
| **STATUS** | Optional with the CHANGE and PRINT subcommands. | CHANGE subcommand—Causes the requested status change to be made to the library member. PRINT subcommand—Causes all AllFusion CA-Panvalet library members with the requested status to be selected for the listing. | The alpha parameter and allowable abbreviations for the CHANGE and PRINT subcommands are:  
- DISABLE  
- ENABLE  
- PROD  
- INACTIVE  
- ACTIVE  
For the PRINT subcommand only, an additional status abbreviation of TEST is allowed. |
<table>
<thead>
<tr>
<th>Keyword</th>
<th>Where Used</th>
<th>Action</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>TSO</td>
<td>Optional with the ADD, RETRIEVE, DISPLAY, and STORE subcommands.</td>
<td>ADD, STORE subcommands—Causes the library member to be added or stored with TSO internal format for BAL, ANSCOBOL, COBOL, FORTRAN, PL/1, JCL, or any ZTYPEs that mimic the mentioned language types, unless the member is currently stored as NOFORMAT. All other language types are automatically stored with AllFusion CA-Panvalet internal format. When the member is added or stored according to TSO internal format, the appropriate sequence file is retained. If the input data set is in non-ascending sequence, the ADD/STORE fails. Use of this parameter allows an override of the system installation option to use TSO internal formatting.</td>
<td></td>
</tr>
<tr>
<td>TSO     (continued )</td>
<td>RETRIEVE and DISPLAY subcommands—Cause members of BAL, ANSCOBOL, COBOL, FORTRAN, PL/1, or JCL language formats to be automatically sequenced in TSO formats for retrieval or display. This applies to any ZTYPEs that mimic the mentioned languages. All other language types and all NOFORMAT types are treated with PANVALET sequencing when applicable.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Keyword</strong></td>
<td><strong>Where Used</strong></td>
<td><strong>Action</strong></td>
<td><strong>Format</strong></td>
</tr>
<tr>
<td>------------</td>
<td>----------------</td>
<td>------------</td>
<td>------------</td>
</tr>
<tr>
<td>Use of this parameter allows an override of the system installation option to use TSO sequencing on all retrievals or displays. With this parameter specified, you can retrieve a maximum of 99,999 records from the library member.</td>
<td>TYPE</td>
<td>Optional with the PRINT subcommand.</td>
<td>Causes all AllFusion CA-Panvalet library members with the requested language type to be selected for the listing.</td>
</tr>
<tr>
<td><strong>Note:</strong> TSO-sequenced members should not be larger than 99,999 physical statements; therefore, retrieves or displays should not go beyond physical statement 99,999. Using PANVALET sequencing allows the retrieval or display beyond physical statement 99,999.</td>
<td>USER</td>
<td>Optional with the CHANGE, ADD, and PRINT subcommands.</td>
<td>CHANGE subcommand—Causes the user code or user security level for the library member to be changed to a new value.</td>
</tr>
<tr>
<td>ADD subcommand—Causes a user code to be established. If omitted, user code is zero.</td>
<td></td>
<td></td>
<td>CHANGE subcommand—If there is a user security level, the user code and user security level, which is a one digit numeric value, are supplied in the order; user code, user security level.</td>
</tr>
</tbody>
</table>
**Keyword Summary**

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Where Used</th>
<th>Action</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRINT subcommand—Causes all AllFusion CA-Panvalet library members with the requested user code or range of user codes to be selected for listing.</td>
<td>PRINT subcommand—You can supply one or two user code values. When you specify one user code value, only the user code specified is selected. When two user code values are specified (separated by a space), all user codes inclusive of the first value and the second value are selected. The listing remains in AllFusion CA-Panvalet name sequence.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VOLUME</td>
<td>Optional with the RETRIEVE subcommand.</td>
<td>Causes the requested member to be written to a specific volume.</td>
<td></td>
</tr>
</tbody>
</table>

**Note:** DATASET has the following default naming conventions:

`userid.panname.type`

Where `userid` is a one- to seven-position ID from the User Profile, `panname` is a one to eight high-order position of the library member’s name, and `type` is a TSO qualifier.

For input operations, you are prompted for the correct qualifier if there are multiple qualifiers for the DSNAME. For output operations, a qualifier is selected according to the AllFusion CA-Panvalet internal format, as follows:

<table>
<thead>
<tr>
<th>AllFusion CA-Panvalet Format</th>
<th>TSO Qualifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>BAL, ALC</td>
<td>ASM</td>
</tr>
<tr>
<td>COBOL, COBOL-72, ANSCOBOL</td>
<td>COBOL</td>
</tr>
<tr>
<td>FORTRAN</td>
<td>FORT</td>
</tr>
<tr>
<td>PL/I, PL/1</td>
<td>PLI</td>
</tr>
<tr>
<td>OBJECT</td>
<td>OBJ</td>
</tr>
<tr>
<td>JCL</td>
<td>CNTL</td>
</tr>
<tr>
<td>AUTOCODER, RPG, DATA, OTHER</td>
<td>DATA</td>
</tr>
</tbody>
</table>
When you supply DATASET as a single level (DSNAME), the following default is used:

userid.DSNAME.type
  | supplied

When you supply DATASET as a qualified name (DSNAME.XTYPE), the following default is used:

userid.DSNAME.XTYPE
  | supplied

When you supply DATASET as a quoted value ('ABSOLUTE'), the system uses the exact value supplied:

ABSOLUTE
This appendix contains sample TSO sessions using the PAN Command Processor subcommands. This appendix also contains a sample TSO session using suppressed subcommands.

This section contains the following topics:

- **RETRIEVE and STORE Subcommands** (see page 56)
- **STORE, CHANGE, COPY, RENAME, and DISPLAY Subcommands** (see page 58)
- **CHANGE, RENAME, and LISTDS Subcommands** (see page 60)
- **CHANGE, ADD, and LISTDS Subcommands** (see page 61)
- **PRINT Subcommand** (see page 63)
- **Suppressed Subcommands** (see page 64)
- **PRINT, LOCK, RETRIEVE, UNLOCK, and STORE Subcommands** (see page 66)
RETRIEVE and STORE Subcommands

<table>
<thead>
<tr>
<th>Code</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logon READY alloc fi(pandd1) da('panvalet') shr1</td>
<td></td>
</tr>
<tr>
<td>READY</td>
<td>You can skip this step if PANDD1 is allocated in the logon process.</td>
</tr>
<tr>
<td>pan ret a2501835v4</td>
<td></td>
</tr>
<tr>
<td>DONE 1488 STMT(S) LEVEL 015</td>
<td></td>
</tr>
<tr>
<td>READY</td>
<td></td>
</tr>
<tr>
<td>edit a2501835 asm old</td>
<td>An AllFusion CA-Panvalet member called A2501835V4 is retrieved to a TSO sequential data set called userid.A2501835.type, where type is a type qualifier developed from the AllFusion CA-Panvalet format.</td>
</tr>
<tr>
<td>.</td>
<td></td>
</tr>
<tr>
<td>.</td>
<td></td>
</tr>
<tr>
<td>.</td>
<td></td>
</tr>
<tr>
<td>Save SAVED end</td>
<td></td>
</tr>
<tr>
<td>READY</td>
<td></td>
</tr>
<tr>
<td>pan store a2501835v4 lev(15)3</td>
<td></td>
</tr>
<tr>
<td>DONE 1535 STMT(S) LEVEL 016</td>
<td></td>
</tr>
<tr>
<td>READY</td>
<td></td>
</tr>
<tr>
<td>pan ret a2501835v4 expand4</td>
<td></td>
</tr>
<tr>
<td>This retrieval creates a copy with the same name as the retrieve request above, but embedded ++INCLUDE statements are expanded. You can skip this step if the data set contains no includes.</td>
<td></td>
</tr>
<tr>
<td>Code</td>
<td>Explanation</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>DATASET A2501835.ASM IS ABOUT TO BE REUSED ...</td>
<td></td>
</tr>
<tr>
<td>* *ENTER CARRIER RETURN TO CONTINUE OR ATTENTION TO SPECIFY NEW COMMAND</td>
<td></td>
</tr>
<tr>
<td>NOTE PV184 V4MACROS EXPANDED DONE 3522 STMT(S) LEVEL 016 READY</td>
<td></td>
</tr>
<tr>
<td>asm a2501835 list test print(a25018835. list)5</td>
<td></td>
</tr>
<tr>
<td>NO STATEMENTS FLAGGED ... READY link ...</td>
<td></td>
</tr>
<tr>
<td>READY test6</td>
<td></td>
</tr>
</tbody>
</table>

Compile the member using an interactive compiler or assembler, or set up a job for use with the SUBMIT subcommand.

At the end of the logon session, you can delete the TSO data sets, since you can retrieve them from the AllFusion CA-Panvalet library at a future session.

* This message is not displayed when retrieving to a PDS.
### STORE, CHANGE, COPY, RENAME, and DISPLAY Subcommands

<table>
<thead>
<tr>
<th>Code</th>
<th>Explanation</th>
</tr>
</thead>
</table>
| \logon...

READY | Once a program is tested and ready to be frozen, retrieve again. |
| pan r a2501835

DONE, 298 STMT(s)

level 018

READY | Then store the data set using the AllFusion CA-Panvalet operand so the TSO sequence numbers are dropped. After the store is completed, the TSO data set is deleted as requested by the DELETE operand. |
| pan st a2501835

lev(18) panv

delete2

DONE, 298 STMT(S)

LEVEL 019

INPUT DELETED

READY | Issue a change request with the status operand to flag the member as production on the AllFusion CA-Panvalet library. |
| pan ch a2501835

stat(prod)3

DONE

READY | Rename the program, if desired, with a production type naming convention. |
| pan rename

a2501835

a2501835v4

DONE

READY | Apply a user comment describing the production program, if desired. |
| pan change

a2501835v4

comment

('inventory control master file update v4 09-13-94')

DONE

READY | Create a test copy for future changes before the production program is deleted from the library. |
| pan copy

a2501835v4

a2501835v6

DONE, 298 STMT(S)

LEVEL 019

READY | |
<table>
<thead>
<tr>
<th>Code</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>pan ch a2501835v4 stat(dis)7 DONE READY</td>
<td>Issue a status change to flag the production program as disabled so the program can be removed to a back-up file in a batch AllFusion CA-Panvalet run.</td>
</tr>
<tr>
<td>pan d a2501835v4 SEQ1(20)SEQ2(30) 8 DATA RECORD 20 DATA RECORD 30 DONE 11 STMT(S) LEVEL 019</td>
<td>Display a portion of a member to the screen for examination.</td>
</tr>
</tbody>
</table>
## CHANGE, RENAME, and LISTDS Subcommands

<table>
<thead>
<tr>
<th>Code</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>logon... READY</td>
<td>Once a program is tested and ready to be placed in production status, issue a retrieve request.</td>
</tr>
<tr>
<td>pan r a25018351 DONE, 4523 STMT(S) LEVEL 015 READY</td>
<td>Rename the AllFusion CA-Panvalet member to its production name.</td>
</tr>
<tr>
<td>pan ren a2501835 a2501835v4 DONE READY</td>
<td>Add the previously retrieved member back under the old test name.</td>
</tr>
<tr>
<td>pan add a2501835 f(asm) u(2742)3 Done, 4523 STMT(S) Level 001 READY</td>
<td>Change the production named member to production, and then disable the member.</td>
</tr>
<tr>
<td>pan ch a2501835v4 stat(prod)4 DONE READY</td>
<td>If desired, you can apply a user comment to the production version to describe the program. Use the PAN LISTDS command to verify the previous changes.</td>
</tr>
<tr>
<td>pan ch a2501835v4 stat(dis) DONE READY</td>
<td></td>
</tr>
<tr>
<td>pan Ld a2501835v4 com5 ('version 4 master file update') NAME A2501835v4 148...</td>
<td></td>
</tr>
<tr>
<td></td>
<td>* VERSION 4 MASTER FILE UPDATE DONE READY</td>
</tr>
</tbody>
</table>
## CHANGE, ADD, and LISTDS Subcommands

### Multiple Libraries

<table>
<thead>
<tr>
<th>Code</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>logon/...</code></td>
<td>To select a member from a production AllFusion CA-Panvalet library and create a test copy on a test AllFusion CA-Panvalet library, first allocate the libraries to different ddnames.</td>
</tr>
<tr>
<td><code>alloc fi(pandd1)</code></td>
<td></td>
</tr>
<tr>
<td><code>da('test.source.lib') shr1</code></td>
<td></td>
</tr>
<tr>
<td><code>READY</code></td>
<td></td>
</tr>
<tr>
<td><code>alloc fi(panprod)</code></td>
<td>Use a PAN LISTDS to inspect the production member, supplying the ddname of the production library.</td>
</tr>
<tr>
<td><code>da('prod.source.lib') shr</code></td>
<td></td>
</tr>
<tr>
<td><code>READY</code></td>
<td></td>
</tr>
<tr>
<td><code>pan Ld xyzv5</code></td>
<td>Retrieve the production member.</td>
</tr>
<tr>
<td><code>dd(panprod) com2</code></td>
<td></td>
</tr>
<tr>
<td><code>NAME LVL...STT</code></td>
<td></td>
</tr>
<tr>
<td><code>XYZV5 33 PDA</code></td>
<td></td>
</tr>
<tr>
<td><code>* VERSION 5 RETAIL FORECAST REPORT</code></td>
<td></td>
</tr>
<tr>
<td><code>DONE</code></td>
<td></td>
</tr>
<tr>
<td><code>READY</code></td>
<td></td>
</tr>
<tr>
<td><code>pan r xyzv5</code></td>
<td></td>
</tr>
<tr>
<td><code>dd(panprod)3</code></td>
<td></td>
</tr>
<tr>
<td><code>DONE, 2852</code></td>
<td></td>
</tr>
<tr>
<td><code>STMT(S) LEVEL 033 READY</code></td>
<td></td>
</tr>
<tr>
<td><code>pan add xyzv6</code></td>
<td>Add the production member to the test library. The DDNAME is not required if you are using PANDD1.</td>
</tr>
<tr>
<td><code>da(xyzv5) f(plt)</code></td>
<td></td>
</tr>
<tr>
<td><code>u(253)4</code></td>
<td></td>
</tr>
<tr>
<td><code>DONE, 2852</code></td>
<td></td>
</tr>
<tr>
<td><code>STMT(S) LEVEL 001 READY</code></td>
<td></td>
</tr>
</tbody>
</table>
### Change, Add, and LISTDS Subcommands

<table>
<thead>
<tr>
<th>Code</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>pan changte xyzv6 com5</td>
<td>Change the user comment, if desired, to reflect the next program being developed.</td>
</tr>
<tr>
<td>('version 6 retail forecast monitor report')</td>
<td></td>
</tr>
<tr>
<td>DONE</td>
<td></td>
</tr>
<tr>
<td>READY</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>pan Ld xyzv66</td>
<td>Check out member's attributes, if desired, using the LISTDS subcommand.</td>
</tr>
<tr>
<td>NAME</td>
<td></td>
</tr>
<tr>
<td>LVL...STT...</td>
<td></td>
</tr>
<tr>
<td>XYZV6 001</td>
<td></td>
</tr>
<tr>
<td>TEA...</td>
<td></td>
</tr>
<tr>
<td>DONE</td>
<td></td>
</tr>
<tr>
<td>READY</td>
<td></td>
</tr>
</tbody>
</table>
**PRINT Subcommand**

<table>
<thead>
<tr>
<th>Code</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>logon...</td>
<td>To prepare for an impending deleted run, invoke the PAN command processor to perform a STATUS...DISABLE subcommand on each data set that is to be deleted in a batch AllFusion CA-Panvalet run.</td>
</tr>
<tr>
<td>pan ch xyz s(Sd)</td>
<td></td>
</tr>
<tr>
<td>done</td>
<td></td>
</tr>
<tr>
<td>ready</td>
<td></td>
</tr>
<tr>
<td>pan ch xyw s(d)</td>
<td></td>
</tr>
<tr>
<td>done</td>
<td></td>
</tr>
<tr>
<td>ready</td>
<td></td>
</tr>
<tr>
<td>pan ch abc s(d)</td>
<td></td>
</tr>
<tr>
<td>done</td>
<td></td>
</tr>
<tr>
<td>ready</td>
<td></td>
</tr>
<tr>
<td>pan print s(d)</td>
<td>A PAN PRINT request can be invoked to list back all data sets that are in disabled status only. The library control code is prompted for in print bypass mode.</td>
</tr>
<tr>
<td>ENTER CONTROL CODE</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NAME</th>
<th>LVL...STT</th>
<th>STT</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABC</td>
<td></td>
<td>TDA</td>
</tr>
<tr>
<td>XYZ</td>
<td></td>
<td>TDI</td>
</tr>
<tr>
<td>35 DATASET(S) SELECTED</td>
<td></td>
<td></td>
</tr>
<tr>
<td>done</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ready</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Suppressed Subcommands

<table>
<thead>
<tr>
<th>Code</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>logon... READY</td>
<td>The RENAME and USER subcommands are suppressed either on the library or in the PAN command processor.</td>
</tr>
<tr>
<td>alloc fi(pand1) da('crw.panvalet ') shr1 READY</td>
<td></td>
</tr>
<tr>
<td>pan LD payrollv5 NAME LVL... PAYROLLV5 017... READY</td>
<td></td>
</tr>
<tr>
<td>pan rename payroll v5 tpayrollv52</td>
<td>If you supply the control operand without the value, the code is prompted for in print bypass mode.</td>
</tr>
<tr>
<td>pan ch tpayrollv5 user(45 1) con(68)3 DONE READY</td>
<td>You can also supply the control code in parentheses. The second value in the USER operand establishes a security level on the data set.</td>
</tr>
<tr>
<td>pan r tpayrollv54</td>
<td>Any reference by name to the protected member must be accompanied by the access code to be successful.</td>
</tr>
</tbody>
</table>

"Primed Subcommands"
<table>
<thead>
<tr>
<th>Code</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>pay rtppayro5 a ENTER ACCESS CODE DONE, 1540 STMT(s) LEVEL 017 READY</td>
<td>If you supply the access operand alone, the value is prompted for in print bypass mode.</td>
</tr>
</tbody>
</table>
# PRINT, LOCK, RETRIEVE, UNLOCK, and STORE Subcommands

<table>
<thead>
<tr>
<th>Code</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>alloc f(pandd1) da('pan.panvalet') shr READY</td>
<td></td>
</tr>
<tr>
<td>pan print user(0) con(0) (NOTE: The control code is always required.)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NAME</th>
<th>LVL</th>
<th>USER</th>
<th>TYPE</th>
<th>STAT</th>
<th>MAINT</th>
<th>ACCESS</th>
<th>BLKS</th>
<th>STATMTS</th>
<th>ACT</th>
<th>AVG %</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMEMBER</td>
<td>2</td>
<td>COB</td>
<td>TAE</td>
<td>011295</td>
<td>021595</td>
<td>1</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>REN</td>
<td>22</td>
<td>1.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMEMBER</td>
<td>10</td>
<td>ASM</td>
<td>PAE</td>
<td>100594</td>
<td>021595</td>
<td>1</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>REN</td>
<td>20</td>
<td>1.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CMEMBER</td>
<td>4</td>
<td>ASM</td>
<td>TAE</td>
<td>120694</td>
<td>021595</td>
<td>2</td>
<td>84</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOC</td>
<td>75</td>
<td>57.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NMEMBER</td>
<td>2</td>
<td>ASM</td>
<td>PAE</td>
<td>100594</td>
<td>021595</td>
<td>1</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>REN</td>
<td>20</td>
<td>1.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PMEMBER</td>
<td>6</td>
<td>ASM</td>
<td>TAE</td>
<td>111094</td>
<td>021595</td>
<td>1</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOC</td>
<td>20</td>
<td>1.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5 DATASET(s) SELECTED

[ FILE VERSION

PANDD1 : V 14.4

DONE

READY]

The PAN PRINT command provides a directory listing. The ACT (last action) field members CMEMBER and PMEMBER show LOC, thus signifying those members are currently locked and cannot be modified.

pan lock amember

DONE

READY2

This command marks AMEMBER as locked. You can fully access this member, but cannot modify it in the AllFusion CA-Panvalet library using any AllFusion CA-Panvalet subcommands.

**Note:** Locking the member in the AllFusion CA-Panvalet library assures you that no other user can modify the member.
<table>
<thead>
<tr>
<th>Code</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>pan ret amember</td>
<td>This command retrieves the CA-Panvalet member AMEMBER to a TSO sequential data set called userid.AMEMBER.type where type is taken from the member type on the AllFusion CA-Panvalet library.</td>
</tr>
<tr>
<td></td>
<td>You can now edit the sequential data set using your TSO editor or download the data to your personal workstation for editing.</td>
</tr>
<tr>
<td></td>
<td>To update the data into the AllFusion CA-Panvalet library as member AMEMBER, you must UNLOCK the member.</td>
</tr>
<tr>
<td>pan unlock amember</td>
<td>This command stores the changed copy of the member back into the AllFusion CA-Panvalet library.</td>
</tr>
</tbody>
</table>
Appendix B: Exit Facilities

An exit facility is provided for the AllFusion CA-Panvalet Option for TSO to accommodate user-written routines in the PAN Command Processor.

This appendix describes how to use user exits for AllFusion CA-Panvalet Option for TSO.

This section contains the following topics:

- PAN Exits (see page 69)
- Linkage Conventions (see page 70)
- How to Use PANEXIT (see page 75)
- Codes Passed Between PANTSOEX and PAN (see page 78)
- Modifying Installation Defaults (see page 81)
- Suppressing Subcommands (see page 81)
- Sequence Numbering and Retrieval (see page 82)
- Defining Work Data Set Attributes (see page 83)
- Setting the Number of Blocks to Be Allocated for Retrieve Data Sets (see page 84)

PAN Exits

The design of the PAN exit facility is similar though not identical to the exit facility designed for the batch AllFusion CA-Panvalet system. The following capabilities are available to the user in the PAN exit structure:

- Inspect the initial command buffer and the Command Processor Parameter List (CPPL) before PAN performs any action on the command
- Inspect all data records read or written
- Modify any or all records read or written
- Delete records on any input or output file
- Allow the user to perform all the input or output using his or her own files
- Inspect all PAN error and informational messages on lines before they are written to the user’s terminal
- Receive control at end-of-job to complete any processing by the user exit
Linkage Conventions

The PAN Command Processor calls a module named PANTSOEX (CSECT name) for each of the following conditions:

- A first pass, which allows the user to examine the Command Processor Parameter List (CPPL) and the command buffer.
- A separate pass for each error or informational message or line created by PAN. The messages are passed before being sent to the user’s terminal with the TPUT macro. Messages created by TSO service routines are not passed to the exit.
- Two passes made for each read during a PAN ADD or STORE command; one pass before the read and one after.
- One pass made before each record written on a PAN RETRIEVE command, with a second pass possible after each write if the user exit so requests.
- An end-of-file pass after reaching an EOF condition on the input file during a PAN ADD or STORE.
- An end-of-job pass. PAN immediately returns to the system after the return from the user exit.

PAN passes two or three arguments, the address of which is in Register 1. All arguments are not present on all passes and contain varying information depending on the nature of the pass. Register 13 contains the address of an 18 fullword save area where the user must appropriately store the registers. Return is accomplished by Register 14. See the example at the end of this appendix.

You must follow this register convention, which is the standard convention used in higher-level languages such as COBOL. Generally, you cannot alter the parameters passed, except where specifically allowed. PAN sets the parameter referred to as IOCODE dependent in the operation being performed. The user can change IOCODE to reflect his needs before returning to PAN.

Parameters

Upon entry to the user exit, Register 1 contains the address of the parameter list. The parameter list is organized according to the request as follows:
First Pass Parameter List

First pass with command buffer, occurs for all PAN subcommands. Register 1 points to a 2 fullword parameter list, each fullword containing an address.

- The first fullword is the address of another fullword in storage that contains the original Register 1 value passed to PAN by the Terminal Monitor (TMP). This original Register 1 is a pointer to the Command Processor Parameter List (CPPL), which consists of four fullword addresses. The first fullword in this list is the address of the command buffer.

- The second fullword is the address of a three byte code field. The first byte (IOCODE) has a value of X'09' on this pass. The second byte is always C'1'. The remaining byte is not used on the pass.
Message Pass Parameter List

Any pass for PAN messages or lines occurs with all PAN subcommands and includes the print lines created during a PAN LISTDS, PRINT, or DISPLAY.

Register 1 points to a 2 fullword parameter list, each fullword containing an address.

- The first fullword is the address of another 2 fullword list containing the length (four bytes) and location (four bytes) of the messages about to be written to the user terminal.
- The second fullword is the address of a three-byte code field. The first byte (IOCODE) has a value of X'10' on this pass. The second byte is always C'1'. The remaining byte is not used on this pass.
### General Pass Parameter List

Any other pass. This can be the reading or writing of data (PAN ADD, STORE, RETRIEVE only), an end-of-file pass for input (PAN ADD and STORE only), or an end-of-job pass (last pass, occurs for all subcommands).

Register 1 points to a 3 fullword parameter list, each fullword containing an address.

- The first fullword contains the address of the record to be read or written. Its length is 80 for an input record and 81 for an output record.

- The second fullword is the address of a three-byte code field. Byte 1 is the IOCODE, byte 2 is the program, and byte 3 is the return request. These codes can have the following values:
  
  - **IOCODE**—going to the exit:
    - C'1' (Before a read)
    - C'2' (After a read)
    - C'5' (Before a write)
  
  - C'6' - After a write when user requested a return:
    - C'7' (EOF on PAN input)
    - C'8' (EOJ)
  
  - **IOCODE**—Returning from the exit, no change to original value—process as normal:
    - C'A' (User-supplied input record)
    - C'B' (Skip current input record)
    - C'G' (User says EOF on input file)
    - C'E' (Write current output record)
    - C'F' (Skip current output record)

  - **PROGRAM**—C'1 (Constant value indicating PAN is executing) You cannot change this value.

  - **RETURN**—Going to the exit:
    - C'N' (No return setting always set by PAN before going to the exit)

  - RETURN (Returning from the exit, no change—process as normal:
    - C'Y' (Set by user to indicate a return pass is desired after an output record is written or skipped. This value is ignored unless processing an output file.)
How to Use PANEXIT

Checks should be made during the first pass ICODE=X'09' to determine the PAN subcommand being requested. Only RETRIEVE, DISPLAY, ADD, or STORE can process an input or output file.

If you want to read or write using your own input or output data files, you must allocate (either by preallocation or using DAIR) and open your own ddnames. PAN attempts to allocate, open, close, and free the PANDD2 and PANDD3 ddnames, as necessary.

You should check for ICODES = C'1', C'2' or C'7' if data input file processing is performed.

IOCODES = C'5', C'6' indicate a data output file is being processed.

PAN passes the EOJ ICODE = C'8' immediately preceding the return to the operating system. If control is returned to PAN, it goes to the end of the job regardless of the ICODE you returned.

If you want to create a log of all terminal I/O, you can trap the command buffer in the first pass ICODE = X'09' and any PAN terminal output with succeeding ICODE = X'10' passes.

The following types of messages are passed to the exit:

- All PAN ERROR and NOTE messages
- The DONE message created at the completion of a subcommand
- All PAN informational (INPUT DELETED) messages
- All headers and directory lines created by the PAN LISTDS and PAN PRINT
- All other data provided by PAN DISPLAY

Changes to ICODE on an X'09', X'10', or X'F8' pass are ignored when PAN receives control back from the exit.

Changes to ICODE are examined only on the return to PAN when processing a data input or output file (during PAN ADDs, STOREs, RETRIEVEs only). The remainder of the discussion on how to use the PAN exits is divided into the two basic modes of handling data, the input mode and output mode.
Input Mode-PAN ADD/STORE

PAN passes an IOCODE of C'5', and optionally C'6', when in output mode. The first fullword in the exit parameter list contains the address of IOAREA, which is 81 bytes in length. The significant portion of the record is positions 2-81. The first character is always ignored.

An IOCODE of C'5' is passed before each write of an output record. You can inspect and alter the input area as desired. This requires no change to IOCODE and a simple return to PAN processes the record as normal. You can place additional records on any output file before or after the current record. You can also skip the current record. The logic used in each of these cases is described below. Assume an IOCODE=C'5' has been passed.

To add records before the current record:
1. Move the current record from the output area to a save area in PANTSOEX.
2. Move the additional record to the output area.
3. Move to a C'Y' (return request) to RETURN.
4. Return to PAN, which writes the record to the output file. Then return to the user exit with an IOCODE=C'6' (user requested return).
5. If more records are to be inserted, move to a C'E' to IOCODE and repeat Steps 2-5. If no more records are desired, move the original record from the save area to the output area, set the IOCODE to C'E', and return.

To add records after the current record:
1. Move a C'Y' (return request) to RETURN.
2. Return to PAN without changing IOCODE. PAN writes the current record and returns to the exit with an IOCODE=C'6'.
3. Move the new record to the output area.
4. Move a C'E' to IOCODE and a C'Y' to RETURN.
5. Repeat steps 3 and 4 until no more records are to be added. Then return without changing the IOCODE.

To skip an output record, skip the current record (delete from output) by setting the IOCODE to C'F' and returning.
Exit Directory Entry (DIRENTRY)

The directory entry currently used by PAN is presented in 0-UP format in DIRENTRY, which is pointed to by the third fullword in the exit parameter list. If the name (positions 1-10) is blank, no entry is available. Position 78 is a ‘I’ whenever PAN is expanding a ++INCLUDE. On input functions, the directory cannot be complete until the function is completed, although the name is usually available.

Linking TSO Exits

During the initial installation, module PAN00 was linked in TSTEP3, which contains the exit handles (CSECT=PAN00). This module, which appears as an unresolved external reference after the normal link-edit in TSTEP4, must be linked with the load module PAN and the user exit. Sample control statements to link PAN with a user exit are shown as follows:

```//TSTEP3                                    - from original link deck to extract
  .                                              PAN00 from the distribution tape
  .
  .
//SYSLMOD DD DSN=&&X...                              .
  .
  .
//TSTEPX EXEC PGM=IEWL...                           .
  .
  .
//PANLMOD DD DSN=&&X...                              - temp load lib with PAN00 output load
  .                                              module
//SYSLMOD DD DSN=                                 - library where current copy of PAN is
  .                                              located
//SYSLIN DD *                                      - must have CSECT name PANTSOEX
  (user object exit)  INCLUDE PANLMOD(PAN00)
  INCLUDE SYSLMOD(PAN)
  ENTRY CPPAN
  NAME PAN(R)
/*
# Codes Passed Between PANTSOEX and PAN

<table>
<thead>
<tr>
<th>Value of IOCODE Passed to PANTSOEX</th>
<th>Valid IOCODE Returned by PANTSOEX</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'09' (first pass a PAN subcommand)</td>
<td>None</td>
</tr>
<tr>
<td>X'10' (a PAN terminal line)</td>
<td>None</td>
</tr>
</tbody>
</table>
| C'1' (before a read)              | No change  
                                    | C'A' (user-supplied input record)  
                                    | C'G' (user says EOF) |
| C'2' (after a read)               | No change  
                                    | C'B' (skip input record)  
                                    | C'G' (user says EOF) |
| C'5' (before a write)             | No change  
                                    | C'E' (user wants to write an output record)  
                                    | C'F' (skip current output record) |
| C'6' (user requested return)      | No change  
                                    | C'E' (user wants to write an output record)  
                                    | C'F' (skip current output record) |
| C'7' (EOF on PAN input file)      | No change  
                                    | C'A' (user-supplied input record)  
                                    | C'B' (skip input record)  
                                    | C'G' (user says EOF) |
| C'8' (end of job)                 | None |

### Value of RETURN

- **C'N'**: No return requested by user—Always set to N before entry to PANTSOEX
- **C'Y'**: Return requested by user—Valid in output mode only
Sample Exit

TSOX     TITLE 'EXIT FOR PAN VERSION M'
PANTSOEX CSECT
R0 EQU 0
R1 EQU 1
R2 EQU 2
R3 EQU 3
R4 EQU 4
R5 EQU 5
R6 EQU 6
R7 EQU 7
R8 EQU 8
R9 EQU 9
R10 EQU 10
R11 EQU 11
R12 EQU 12
R13 EQU 13
R14 EQU 14
R15 EQU 15
***
USING *,15
STM   14,12,12(13)
LR    12,13
LA    13,SAVEAREA
ST    12,4(13)
ST    13,8(12)
DROP  15
BALR  12,0
USING *,12
***
L     R6,0(1)      A(IOAREA)
L     R7,4(1)        A(CODES)
L     R8,8(1)        A(DIRECTRY)
USING IOAREA,6
USING CODES,7
CLI   IOCODE,C'8'  EOJ?
BE    EOJ
CLI   IOCODE,X'09'  CMD?
BE    COMMAND
CLI   IOCODE,X'10'  MSG?
BE    MSG
RETTOPAN EQU *
L     13,4(13)
LM    14,12,12(13)
RR    14
***
COMMAND EQU *
L     R9,0(R6)
L     R9,0(R9)
User Exit Notes

- Thoroughly test all user exits on a test library before putting them to use on a production AllFusion CA-Panvalet library.
- Replicate problems with PAN that involve a user exit without the exit, if possible. This might expedite finding a solution to the problem and eliminate the possibility of the user exit causing the problem.
- Whenever the user exit returns an invalid IOCDE, PAN ignores it and processes as though the IOCDE did not change.
- A loop between PANTSOEX and PAN might occur if you always set RETURN to C'Y' on an output request. PAN always sets RETURN to C'N' upon entry to PANTSOEX.
Modifying Installation Defaults

You can use an SMP/E USERMOD to customize the PAN Command Processor. Customization includes the suppression of subcommands, setting sequence number and retrieval options, and defining work data set attributes.

Suppressing Subcommands

The PAN Command Processor recognizes suppressed subcommands occurring on the library and performs the subcommand only if you supply the correct control code. With the PAN processor, you can leave a subcommand unprotected in the batch system and suppress it from normal use. You can override the suppressed subcommand in the PAN processor with the appropriate control code.

To suppress a PAN subcommand, prepare a code for location X'0022' of the PANTSO CSECT in the PAN module according to the following scheme:

- XFFFF  all allowed (default)
- X'8000'  add allowed
- X'4000'  copy allowed
- X'2000'  level allowed
- X'1000'  rename allowed
- X'0800'  retrieve/display allowed
- X'0400'  status allowed
- X'0200'  store allowed
- X'0100'  user allowed
- X'0080'  comment allowed
- X'0040'  listds allowed
- X'0020'  lock allowed
- X'0010'  unlock allowed

Examples

- REP 0022 FFFF—Allows all PAN subcommands to be used without a control code.
- REP 0022 FAFF—Suppresses only the STATUS and USER subcommands.
Sequence Numbering and Retrieval

Formatting Options for Retrieve or Display

You can retrieve or display AllFusion CA-Panvalet members with the usual AllFusion CA-Panvalet sequencing (sequence by ones) or with TSO sequencing (sequence by tens). You can establish either of these parameters as the installation default. You can then override the default, if desired, by using the opposite parameter. If TSO is the default when retrieving or displaying a member for which TSO sequencing is not supported, normal AllFusion CA-Panvalet sequencing occurs and no error is reported.

You can perform retrievals or displays with or without expansion of embedded ++INCLUDEs by using EXPAND or NOEXPAND parameters. You can choose either of these parameters as an installation default with override possible by using the opposite parameter.

To establish the retrieval options, prepare a two-byte hexadecimal field to be placed in position X'0016' of PANTSO as follows:

X'D7D5'  'PN' - CA-PANVALET/NOEXPAND
X'E3D5'  'TN' - TSO/NOEXPAND
X'D7C5'  'PE' - CA-PANVALET/EXPAND
X'E3C5'  'TE' - TSO/EXPAND

Example

REP 0016 E3C5 causes all data sets to be retrieved with TSO compatible sequence numbers (when the language format permits) and expansion of INCLUDEs with header/trailer comments.

Note: Keep TSO sequenced members smaller than 99,999 physical statements; therefore, retrieves or displays should not go beyond physical statement 99,999. Using AllFusion CA-Panvalet sequencing allows the retrieval or display beyond physical statement 99,999.
Formatting Options for Add or Store

When the language format permits, records written to the AllFusion CA-Panvalet library can be automatically stored in TSO internal format (retaining sequence numbers) by establishing the TSO operand as the installation default. If you want normal CALLFusion A--Panvalet internal formatting (dropping sequence numbers), you should choose PANVALET as the default.

To establish this formatting option, prepare a two-byte code to be placed in PANTSO as follows:

```
REP 0018 D740 'P' - CA-Panvalet or
REP 0018 E340 'T' - TSO
```

**Note:** Keep TSO sequenced members smaller than 99,999 physical statements; therefore, retrieves or displays should not go beyond physical statement 99,999. Using AllFusion CA-Panvalet sequencing allows the retrieval or display beyond physical statement 99,999.

Defining Work Data Set Attributes

Deleting Input Data Set, Updating Date of Last Access

Normally, the input data set used on a PAN ADD or PAN STORE request is kept after the subcommand is completed. If desired, you can establish an installation option to automatically delete and uncatalog the data set after a successful ADD or STORE is performed. The DELETE subcommand is performed only on sequential data sets, not PDSs or PDS members. The installation option, whether to keep or delete, can be overridden at execution time by using the DELETE or KEEP operand with the ADD or STORE subcommands.

The delete option is selected with the Updating Date of Last Access Option. If you specify N (NO), no ENQ or reserve is performed on the AllFusion CA-Panvalet library, and the date of last access is not updated. If you specify Y (YES), the date of last access on a PAN R or PAN D is updated. Prepare a two-byte code to be placed in the PANTSO module for the option combination as follows:

```
REP 0014 D2D5 'K' - KEEP 'N' - NO UPDATE
REP 0014 C4D5 'D' - DELETE 'N' - NO UPDATE
REP 0014 D2E8 'K' - KEEP 'Y' - YES UPDATE
REP 0014 C4E8 'D' - DELETE 'Y' - YES UPDATE
```
Setting Block Size for Retrieve Data Sets

You can specify the desired block size for the TSO data set created with PAN RETRIEVE as the installation standard by modifying the TSO#BLK option of the PVOPT macro. The PVOPT default is block size 3200. For example, TSO#BLK is set to 40 (40 x 80=3200). For more information, see the appendix "PVOPT Macro USERMODs."

You cannot alter this default block size at execution time. However, if you require a different block size, you can preallocate the data set with the block size before invoking the PAN RETRIEVE or perform a TSO COPY after the retrieve with a new block size.

Setting the Number of Blocks to Be Allocated for Retrieve Data Sets

The primary and secondary number of blocks allocated for the data set (created with PAN RETRIEVE) can be specified as the installation standard by preparing a six-byte character format replacement for the primary and secondary number of blocks as follows:

REP  0002C        F0F0,F0F1,F0F0     '000100' Primary Blocks
REP  00032        F0F0,F0F0,F5F0     '000050' Secondary Blocks

When blocks are allocated, the exact number may not be exactly as requested, due to the propagation of blocks when filling out the remainder of a track. This occurs when the number of allocated blocks falls short of the track capacity.

After installation, the next procedure is the initialization of the AllFusion CA-Panvalet library. See the System Management Guide for more information.
Index

A
ACCESS • 19, 32, 34, 35, 36, 39, 41, 42, 43
ACCESS keyword • 19, 32, 34, 35, 36, 39, 41, 42, 43
accessing multiple libraries • 27
ADD • 15, 25, 31
ADD subcommand • 15, 16, 18, 31, 43, 61, 83
ADD/STORE • 83
adding data set • 25
AllFusion CA-Panvalet library file • 11
AllFusion CA-Panvalet sequencing • 82
allocating ddname • 27
arguments passed by PAN exit • 70
attention interrupt • 30

B
batch AllFusion CA-Panvalet programs • 29
batch mode update • 23
blocks allocated for retrieve data sets • 84
blocksize • 84

C
capabilities • 69
CHANGE • 18, 31
CHANGE subcommand • 18, 19, 20, 31, 58, 60, 61
changing blocksize • 84
command processor • 27
COMMENT • 19, 35, 37, 43
COMMENT keyword • 19, 35, 37, 43
document record • 23
CONTROL • 16, 19, 32, 34, 35, 36, 37, 39, 41, 42, 43
CONTROL keyword • 16, 19, 32, 34, 35, 36, 37, 39, 41, 42, 43
date of last access • 83
copy • 20, 31
COPY • 31
COPY subcommand • 20, 31, 32, 58, 60
creating TSO data set • 84

D
data handling, PANEXIT • 75
DATASET • 16, 39, 41, 43
DATASET keyword • 16, 39, 41, 43
date of last access • 83
DDNAME • 16, 19, 32, 34, 35, 36, 37, 39, 41, 42, 43
DDNAME keyword • 16, 19, 32, 34, 35, 36, 37, 39, 41, 42, 43
ddname, allocating • 27
Default naming conventions • 43
defining work data set attributes • 83
DELETE • 16, 41, 43
DELETE keyword • 16, 41, 43
deleting • 83
deleting input data set • 83
descriptions • 15
DIRENTRY • 77
DISPLAY • 20, 32
DISPLAY subcommand • 20, 32, 34, 58, 60

E
elements • 18, 20, 32, 36, 38, 43
executing AllFusion CA-Panvalet under TSO • 27
exit directory entry, DIRENTRY • 77
EXPAND • 34, 39, 43
EXPAND keyword • 34, 39, 43, 82
expanding ++INCLUDEs • 25, 82

F
file requirements and options • 10
flow diagram • 10
FORMAT • 16, 43
FORMAT keyword • 16, 43
formatting option • 23
formatting options • 82, 83
function • 15, 18, 20, 21, 22

I
IKJDAIR • 30
IKJDAIR routine • 30
IKJDFT • 30
IKJDFT routine • 30
IKJPARS • 30
IKJPARS routine • 30
IKJSN • 30
IKJSN routine • 30
initializing • 11
initializing AllFusion CA-Panvalet library file • 11
input data handling, PANEXIT • 75
input data set • 83
input work file • 13
input/output flow diagram • 10
interactive functions • 30
interface between TSO/Batch processing • 23
internal organization • 30
invoking • 27
IOCODE parameter • 70

K
KEEP • 16, 41
KEEP keyword • 16, 41
keyword summary • 43
keywords • 16, 19, 32, 34, 35, 36, 37, 39, 41, 42

L
LEVEL • 19, 41
LEVEL keyword • 19, 41
linkage conventions, PAN exits • 70
linking TSO exits • 77
LISTDS • 20, 35
LISTDS subcommand • 20, 35, 36, 61
LOCK • 21, 36
LOCK subcommand • 21, 36, 66

M
member names • 16, 19, 32, 34, 35, 36, 37, 39, 41, 42
modifying installation defaults • 81
multiple libraries • 11, 27, 61

N
NAME • 37
NAME keyword • 37, 43
naming convention, work file • 12
NOEXPAND • 34, 39
NOEXPAND keyword • 34, 39, 43, 82
NOFORMAT • 16
NOFORMAT keyword • 16

O
output data handling, PANEXIT • 75
output mode, PAN RETRIEVE/DISPLAY • 76
output work file • 12

P
PAN attention interrupt • 30
PAN attention interrupt routine • 30
PAN command processor • 27, 30
PAN Command Processor • 81
PAN command processor subcommand • 15, 18, 20,
21, 22, 25, 31, 32, 35, 36, 37, 38, 40, 42
PAN D • 83
PAN exits • 69, 70
PAN R • 83
PAN subcommands • 15
PAN subcommands syntax • 31, 32, 35, 36, 37, 38,
40, 42
PAN#4 • 29
PANDD1 • 11
PANDD1, library file • 11
PANDD2, output work file • 12
PANDD3, input work file • 13
PANEXIT, using • 75
PANTSOEX module • 70
PANVALET • 16, 34, 39, 41
PANVALET keyword • 16, 34, 39, 41, 43
parameters • 70
parameters, PAN exits • 70
primary • 84
PRINT • 21, 37
PRINT NAME • 37
PRINT subcommand • 21, 37, 63, 66
processing features • 23, 25

R
RECLN • 41
RECLN keyword • 41
register convention • 70
RENAME • 21, 38
RENAME subcommand • 21, 38, 58, 60
resequencing program • 23
RETRIEVE • 21, 38
retrieve data sets • 84
RETRIEVE subcommand • 12, 21, 38, 39, 56, 66
RETRIEVE/DISPLAY • 82
routines • 30

S
sample TSO session • 56, 58, 60, 61, 63, 66
secondary • 84
selecting data • 12
SEQ1 • 34
SEQ1 keyword of DISPLAY subcommand • 34
SEQ2 • 34
SEQ2 keyword of DISPLAY subcommand • 34
sequence numbering and retrieval options • 82
sequencing • 82
special processing features • 25
STATUS • 19, 37
STATUS keyword • 19, 37, 43
STORE • 21, 25, 40
STORE subcommand • 21, 40, 41, 56, 58, 66, 83
storing data set • 25
subcommands • 31, 32, 35, 36, 37, 38, 40, 42, 81
summary • 43
suppressed subcommands • 81
suppressed subcommands, sample TSO session • 64
suppressing • 81
suppressing subcommands • 81
syntax • 27, 31, 32, 35, 36, 37, 38, 40, 42

T
TSO • 16, 34, 39, 41
TSO batch processing interface • 23
TSO data set • 84
TSO exits, linking • 77
TSO keyword • 16, 34, 39, 41
TSO sequencing • 82
TYPE • 37
TYPE keyword • 37

U
UNLOCK • 22, 42
UNLOCK subcommand • 22, 42, 66
updating • 83
updating date of last access • 83
USER • 16, 19, 37
user exit notes • 80
USER keyword • 16, 19, 37
using PANEXIT • 75

V
VOLUME • 39
VOLUME keyword • 39