PDSMAN Introduction

Bruce Koss
Wells Fargo

Monday, August 10, 2015
3:15 PM - 4:15 PM
17410

www.SHARE.org

#SHAREorg

PDSMAN

- Provides solutions to a wide range of PDS and PDSE issues
- Features
  - FASTCOPY replacement for IEBCOPY
  - Dynamic Library Space Reuse
  - Library Member Archiving and Recovery
  - EZYEDIT ISPF Productivity Platform and User Productivity Tools
  - Library Look-Aside (LLA) Extensions
  - PDSMAN Performance Options
  - Auditing and Comparison Solutions
- The next few slides provide a general high level overview of PDSMAN and how it works

Agenda

Topic	Description
History	Past events
Features	Overview of several features
Customization	Initialization parameter deck (PDSMINIT)
Started Tasks	Address spaces, files and commands
Links	PDSMAN and other slides decks
Conclusion	Final comments

Attachments ( )

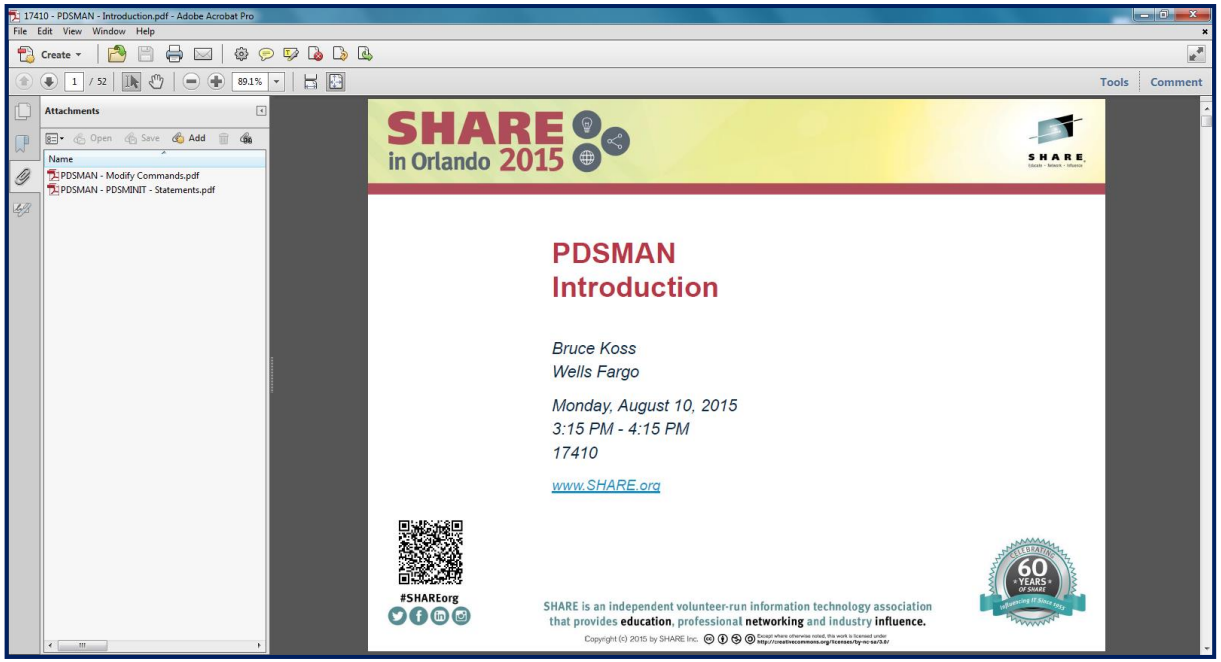
- The following attachments are included in this document:

File	Type
PDSMAN – Modify Commands	PDF
PDSMAN – PDSMINIT – Statements	PDF

Attachments (📎)

- Invoke Adobe Reader and click the 'Attachments' button on the left side of the navigational panel of the main Reader window

Click



Complete your session evaluations online at www.SHARE.org/Orlando-Eval

Disclaimers

- Must use Adobe Reader to view attachments
  - () signifies attachment
- May need to adjust the Adobe Reader zoom value to view text
- Commands and screens may be slightly different depending on the operating system's version and release level
  - These commands and screen prints were executed on
    - PDSMAN/EZYEDIT V7.70 and z/OS V2R1
- Security may prevent access to panels or executing commands
- For more detailed information, please reference the following manual:
  - CA PDSMAN PDS Library Management
    - Administrator Guide

History

Overview

- The next few slides show the history of PDSMAN and its competitor Dataset Commander

PDSMAN – History

Year	Comments
1979	Created by Ian Cairns while working for Software Design Associates (SDA), Australia • Written in Assembler • Software Design Associates
1981	Goal Systems bought the marketing rights • Maintenance and enhancement responsibilities shared between Goal Systems and SDA
1992	Legent acquired Goal Systems
1995	Computer Associates purchased Legent
2014	Computer Associates acquired all rights to PDSMAN • Currently own all of the source code • The same CA team continues to be responsible for PDSMAN (over 80 combined years of experience with the product)

Dataset Commander – History

Year	Comments
1993	Originally released as SPIFFY by Isogon in New York, NY
2005	IBM purchased Isogon • Renamed SPIFFY to ISPF Productivity Tool
2013	IBM renamed it again to Dataset Commander
Today	Links • IBM Dataset Commander • IBM Dataset Commander – Users Guide • IBM Dataset Commander – Installation and Customization Guide

Comments

- PDSMAN/EZYEDIT was created over 36 years ago (in 1979)
  - Dataset Commander has been around for 23 years (since 1993)
- PDSMAN/EZYEDIT has never changed its name
  - Dataset Commander has changed its name three times
- PDSMAN/EZYEDIT has 13 manuals
  - Dataset Commander has 3 manuals
- Every PDSMAN/EZYEDIT release introduces new features and commands which are easy to understand, implement and use

Features

FASTCOPY

- FASTCOPY is a high-speed, transparent replacement for IBM's IEBCOPY utility
- FASTCOPY transparently replaces IEBCOPY and provides both functional and performance enhancements including:
  - Compatible to existing IEBCOPY commands and functions
  - New functionality and capabilities not available in IEBCOPY
  - Improved performance, including faster execution times, reduced CPU usage and fewer I/O operations
  - An integrated volume or dataset pattern driver for library compress and space management functions
- No JCL or operational changes are required to implement the facility
  - Can be enabled or disabled on a global basis, by job name or user ID, or at the individual job step level

Dynamically Reuse PDS Library Space

- Dramatically reduces or eliminates the need to compress PDS datasets
  - Preventing an out-of-space condition or Sx37 abend
    - Occurs when PDS members are deleted or updated as the space occupied by the member is not reclaimed for future use
- Dynamically reuses the dead PDS space when a member is deleted or updated
  - The space occupied by a deleted or updated member is recorded and is made available for immediate reuse
- It is transparent and requires no JCL changes or modifications
  - Special I/O operations are used to minimize overhead and the fail-safe process always maintains the integrity of the data

EZYEDIT

- Provides a general display used to process and save a variety of different datasets, DASD volumes and TSO commands
- Provides numerous commands, capabilities and productivity tools that are not available in ISPF
- Easy to learn for both novice and experienced ISPF users
- Has a comprehensive tutorial including point-and-shoot help and simplified command selection
- Only has three main panels: Selection, Dataset List and Member List

Productivity Tools

- String Scan and Replace
  - Scans and optionally replaces character or hexadecimal strings in all or selected members of a library or library concatenation, or in sequential datasets
- Duplicate Member Reporting
  - Processes the directories of specified library concatenations and reports the presence of members that exist in more than one library
  - The utility can also be used to scan library concatenations to locate all occurrences of specified members
- Library Space Monitoring
  - Produces warning messages when the space usage of selected libraries exceeds defined thresholds
  - This advance warning allows corrective action to be taken before a library fills and causes disruption of critical systems

Productivity Tools

- Library Directory Monitoring
  - Monitors partitioned dataset directories and issues error or warning messages when the amount of directory space being used exceeds established thresholds
  - This advance warning allows corrective action to be taken before the directory fills

- Descriptive Member Titles
  - Used to maintain meaningful titles for library members
  - Titles belonging to library members are maintained in a *title member* within the same library and can be viewed online using EZYEDIT or reported using a batch job
  - Up to ten 60-character lines of descriptive title information can be maintained for each member

Productivity Tools

- Library Empty and Space Release
  - Used to delete all library members in a single operation
  - It can also be used to release unused secondary extents
  - Is faster and more convenient than scratching and reallocating the library because the space and DCB parameters do not need to be re-specified

- Library Map and Analysis
  - Validates and maps partitioned datasets checking for errors, providing diagnostic information and allowing for corrective action to be taken
  - Critical installation datasets can be validated on a regular basis to ensure errors are detected at the earliest possible time

Library Look-Aside

- LLA Auto-Update
  - Eliminates the need for manual LLA intervention and the problem of accessing out-of-date members by updating LLA at the time the member update takes place
- LLA Synchronization Management
  - Performs cache synchronization checking, reporting and automatic resynchronization on a timed basis for specified LLA datasets
- Extended LLA Operator Commands
  - Quick and easy way to modify the LLA environment
  - LLA libraries can be added, removed, updated or have their management mode changed directly from the console using commands such as F LLA, ADD=dsn
  - No longer a need to edit PARMLIB members or learn LLA control statement syntax

Customization

Overview

- PDSMAN features are activated using an initialization parameter deck
- The parameter deck is a PDS member called PDSMINIT
- Features are turned on by coding initialization control statements

PDSMINIT

- The following is an example of coding initialization control statements:

```
*****************************************************************************
*     P D S M A N    I N I T I A L I Z A T I O N    P A R A M E T E R S     *
*****************************************************************************
$BSI      LINKLIST=Y                    /* Linklist Access Reporting         */
          JOBLIB=100                    /* Steplib Acess Recording           */
          LPA=Y                         /* LPA Access Recording              */
          FASTSTOW=Y                    /* STOW Performance Enhancement      */
*****************************************************************************
$ACCESS   LIB=TEST.LOAD    REF=Y        /* Last Reference Date Recording     */
$ACCESS   LIB=PROD.-       MOD=L        /* SMF Logging of access to          */
                           MEM=BW-      /*   all members named 'BW.....'     */
*****************************************************************************
$UPDATE   LIB=TEST.LOAD    CONTROL=Y    /* Member Control Info Recording     */
                           PSR=Y        /* Librasry Space Reuse              */
$UPDATE   LIB=PROD.SOURCE  VERSIONS=2   /* Maintain 2 versions of members    */
                           JOURNAL=Y    /* Up-to-the-minute Recovery         */
$UPDATE   LIB=PROD.CTL     USER=FPRWA   /* Userid FPRWA                      */
                           MODE=A       /*   Allowed to update               */
                           MEM=BW-      /*   All members named 'BW.....'     */
$UPDATE   LIB=PROD.CTL     MODE=P       /* Other users may not update        */
                           MEM=BW-      /*   All members named 'BW.....'     */
*****************************************************************************
$EZYEDIT ACBPREFIX=EZYEDIT              /* For EZYEDIT Remote Printer        */
          HSMPROMPT=YHK                 /* Prompt for recall by HSM          */
```

PDSMINIT

- The following is an example of coding initialization control statements:

```
*************************************************************************
$LLA     LIB=-           UPDATELLA=Y    /* Automated Update LLA and          */
                         NOTIFY=Y       /*   Also issue console messages     */
*************************************************************************
$IEBCOPY NAME=IEBCOPY    FASTCOPY=Y     /* FastCopy replaces IEBCOPY         */
*************************************************************************
$MONITOR LIB=-           CLASS=VALIDATE /* Monitor Library Validation        */
                         HISTORY=S      /* Record History to SMF             */
*************************************************************************
$UTILITY NAME=IEBUPDTE   LIBDD=SYSUT2   /* Intercept updates via             */
                         RPTDD=SYSPRINT /*  IEBUPDTE                       */
*************************************************************************
```

Control Statements ()

- The following are the initialization control statements:

Statement	Description
$ACCESS	Defines rules and options associated with reading or accessing library members
$BSI	Defines global PDSMAN processing options
$COMPILER	Defines names of compilers that will be front-ended by PDSMAN
$DATABASE	Defines the name of the PDSMAN database and associated processing options
$DYNBLDL	Defines libraries or step names to take part in Dynamic BLDL
$EZYCMD	Specifies commands to be dynamically added to the ISPF system command table
$EZYEDIT	Specifies global parameters for the EZYEDIT facility
$EZYICD	Defines ISPF Select service keywords for ISPF initial commands
$IEBCOPY	Defines names under which the IEBCOPY utility can be invoked and PDSMAN processing options for IEBCOPY
$JOURNAL	Defines libraries for library update journaling
$LLA	Defines libraries and options for the LLA/Extensions facility
$LLASYNC	Defines libraries and options for the LLA Synchronization Management facility

Control Statements ()

- The following are the initialization control statements:

Statement	Description
$MISC	Defines miscellaneous variables and global processing options
$MIXED	Defines mixed libraries (those libraries which distinguish between test and production members)
$MONITOR	Defines rules and options related to the monitoring component of the PDSMAN Partitioned Resource Management System
$PFO	Defines JOBLIB/STEPLIB libraries for which Program Fetch Optimization is required
$SECURITY	Defines options related to PDSMAN security
$SYSID	Defines the system IDs for which the rules following this control statement are effective
$UPDATE	Defines rules and options associated with writing or updating library members
$USESTATS	Defines usage statistics recording
$UTILITY	Defines names of system and user-written library update utilities that will be front-ended by PDSMAN

Control Statements – Syntax

- Syntax
  - Free-form
  - Only columns 1 to 72 can be used
  - Only one control statement per line
  - Parameters may be separated by commas or blanks
  - May be specified on multiple lines requiring no continuation characters
    - Individual parameters and their values must be specified on the same line
      - Example: LIB=PROD.SOURCE

- Comments
  - An asterisk (*) placed in front of a control statement is treated as a comment
  - Control statements preceded with a slash followed by an asterisk (/*) will cause all remaining data to be ignored

Control Statements – Wild Cards

- Wild card characters can be used in the parameters
  - The plus sign (+)
    - Represents a single character
  - The minus sign (-)
    - Represents a string of characters

- These characters can be changed using the PATTERN1 and PATTERN2 parameters in the $BSI control statement
  - If changed, the $BSI specification must be coded prior to using any of the wild card characters

Control Statements – Sequence

- When coding multiple control statements, code the most specific ones first, followed by the less specific ones

- The following example activates Library Space Reuse (PSR=Y) on particular datasets while leaving it off for one library (USER.PRODLIB):

```
$UPDATE LIB=USER.PRODLIB
        PSR=N
$UPDATE LIB=USER.-
        PSR=Y
```

- If these control statements were reversed, Library Space Reuse would be in effect for all libraries that begin with USER, including USER.PRODLIB

Control Statements – Syntax Check

- PDSM00
  - A batch utility that checks the syntax of PDSMAN control statements
  - Only the control statement syntax is verified
  - There is no checking of the logical correctness of the statements, product authorization, and so on

- JCL

```
//PDSM00 EXEC PGM=PDSM00,PARM='TEST'
//PDSMRPT DD SYSOUT=class
//PDSMINIT DD DISP=SHR,DSN=dataset
```

Control Statements – PDSM34

- PDSM34
  - A batch utility used to report the $ACCESS and $UPDATE control statements in effect for a given dataset
  - The active control statements in common storage are scanned and all statements matching the specified dataset name are reported

- JCL

```
//PDSM34 EXEC PGM=PDSM34,PARM='dataset'
//PDSMRPT DD SYSOUT=class
```

Directive Statements – #INCLUDE

- Rule directives are used to add flexibility on how to specify and manage PDSMAN control statements

- The #INCLUDE directive allows rules from sources other than the PDSMINIT DD statement in the PDSMAN address space JCL to be dynamically included in the rule input stream
  - Increasing flexibility by allowing to indirectly specify PDSMAN control statement locations

- Syntax:

  #INCLUDE DD=ddname or LIB=dataset[(member)][/volume]

Directive Statements – #INCLUDE

- Only one of the following parameters can be specified:

Parameter	Description
DD	The name of a DD statement in the PDSMAN address space JCL that specifies a PDS member, sequential dataset or concatenation of rule sources containing PDSMAN control statements
LIB	The name of a PDS member or sequential dataset containing PDSMAN control statements • The volume specification allows un-cataloged datasets to be used.

- Examples:
  - #INCLUDE DD=PDSMTEST
  - #INCLUDE LIB=PDSMAN.PARMLIB(DEMORULE)
  - #INCLUDE LIB=PDSMAN.SEQ.RULES/MYVOL1

PDSMAN

- The PDSMAN started task reads the initialization parameters at startup and when the product is re-initialized

- The individual control statements are decoded and stored in common storage for later use by the started task

- The initialization parameter deck can be reloaded by executing the following MVS modify command:

  ➔ F PDSMAN,NEWRULES

  Or

  ➔ F PDSMAN,RELOAD

- It is not necessary to IPL or stop/start PDSMAN in order to activate a new set of initialization control statements

Suggestions

- Implement the features that will provide the most benefit with the least amount of work first

- The following features are good candidates:
  - The EZYEDIT facility, an ISPF-based productivity platform
  - The Productivity Tools facility
  - The FASTCOPY facility, a high-speed replacement for the IEBCOPY utility

- Implementation
  - Turn on the features for a specific set of libraries (or users)
  - Perform testing
  - Then, turn on the features to the general population

Started Tasks

CA Common Services

- PDSMAN uses the following CA Common Services:
  - CAIRIM (Resource Initialization Manager)
  - CA License Management Program (LMP)
  - CA Health Checker Common Service

- LMP Codes

LMP Code	PDSMAN Features
GC	All PDSMAN Features
FY	LLA/Extensions and Performance Options
FZ	Auditing and Comparison Facilities
F2	FastCopy and Library Space Reuse
F3	Member Archiving and Recovery
F6	EZYEDIT and Productivity Tool Facilities

Started Tasks

- The started tasks (address spaces) are as follows:
  - PDSMAN – Main started task
    - Uses the initialization parameter deck (PDSMINIT)
    - Uses a sequential communication file
    - Activates and executes sub-tasks
  - PDSMDB – Database Server Control Task
    - Uses a VSAM KSDS file

PDSMAN

- The PDSMAN started task must be active before any features can be used

- The started task is usually in a wait state
  - Becomes active when timer or subtask driven processing is required
  - Or when a command is entered from the operator's console

- Start and stop
  - ➔ S PDSMAN
  - ➔ P PDSMAN

PDSMAN – Subtasks

- The following subtasks execute within the PDSMAN started task:

Sub-task	Description
PDSMDB	Database Server Control Task
PDSMPMON	Partitioned Library Monitoring
PDSMQMDB	Database Recording Queue Manager
PDSMMCPU	Cross-System Communications
PDSMLMON	Library Look-Aside Monitoring

PDSMAN – Communication File

- The PDSMAN started task uses a sequential file to communicate events to other PDSMAN started tasks executing on different lpars within the sysplex
  - This dataset needs to be accessible to all PDSMAN started tasks executing in the sysplex
  - The file is called the Cross-System Communication facility (PDSMMCPU)

- The following events are logged in this file:
  - Updates to LLA and Dynamic BLDL libraries
  - Extended LLA Operator Commands
  - Partitioned Resource Management messages
  - Other PDSMAN and system-related events

PDSMDB

- The PDSMDB started task is the Database Server Control Task
  - Records information on partitioned datasets
  - Introduced in R7.5
  - This feature is mandatory with release 7.7 and higher

- To start, stop and query the PDSMDB started task
  - Issue operator commands to the PDSMAN started task
  - This task is automatically started and stopped by the PDSMAN started task
    - There is no need to manage it separately

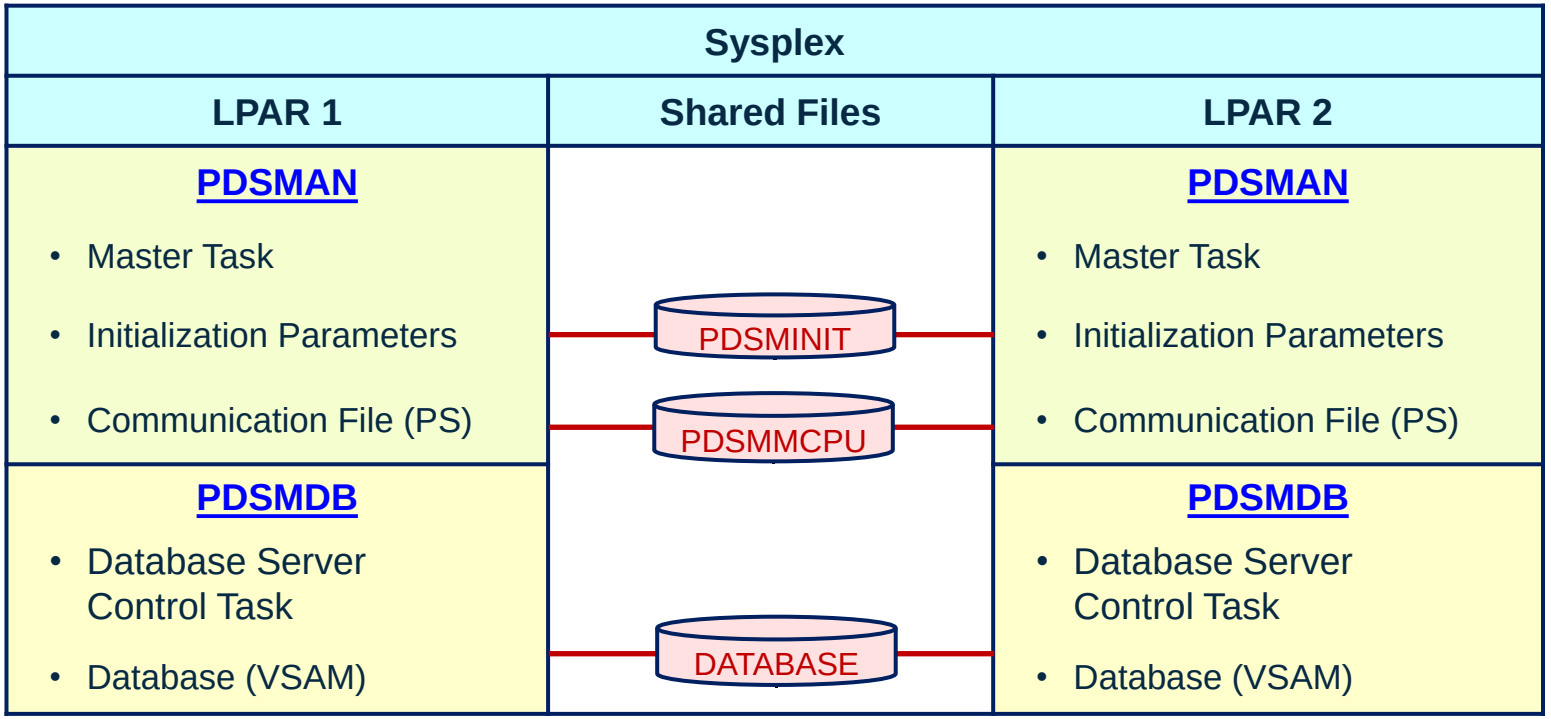
PDSMDB – Database

- The database is a VSAM key-sequenced dataset (KSDS)

- The VSAM file is normally shared and used by all PDSMDB started tasks within the sysplex
  - Allows users access to the most up to date information recorded
  - The VSAM file is defined with SHAREOPTIONS=(3,3)
  - PDSMAN accomplishes its own locking and serialization

Diagram

- The following diagram shows a sysplex with two lpars executing the PDSMAN and PDSMDB started tasks sharing files



F PDSMAN,*command* ()

- The following are the PDSMAN modify commands:

Command	Description
NEWRULES	Re-initializes PDSMAN to activate new initialization control statements
RELOAD	Alias for NEWRULES
STOP	Stop the PDSMAN address space
STATUS	Displays the status of major PDSMAN address space components
DETAIL	Displays detailed information regarding the PDSMAN address space
HELP	Displays important or frequently used PDSMAN address space commands
?	Alias for HELP
STARTMCPU	Starts the PDSMAN Cross-System Communication subtask
STOPMCPU	Stops the PDSMAN Cross-System Communication subtask
NEWMCPU	Bounces the PDSMAN Cross-System Communication subtask
MCPUNET	Displays the current cross-system communications environment status
HANDSHAKE	Verifies the status of all systems in the cross-system environment
XSYSLIBS	Shows the names of all libraries involved in cross-system communication
XSYSLIBS	Shows the names of all libraries involved in cross-system communication

F PDSMAN,*command* ()

- The following are the PDSMAN modify commands:

Command	Description
MCPUSTATUS	Displays the current status of the PDSMMCPU subtask
MCPUINFO	Displays status and other information for the PDSMMCPU subtask
MCPUDETAIL	Displays detailed information about the PDSMMCPU subtask
MCPUPERFORM	Displays performance information about the PDSMMCPU subtask
STARTLMON	Starts the PDSMAN Library Look-Aside (LLA) Monitoring subtask
STOPLMON	Stops the PDSMAN Library Look-Aside (LLA) Monitoring subtask
NEWLMON	Bounces the PDSMAN Library Look-Aside (LLA) Monitoring subtask
LMONSTATUS	Displays the current status of the PDSMLMON subtask
LMONINFO	Displays status and other information for the PDSMLMON subtask
LMONDETAIL	Displays detailed information about the PDSMLMON subtask
LMONPERFORM	Displays performance information about the PDSMLMON subtask
STARTPMON	Starts the PDSMAN Partitioned Resource Monitoring subtask
STOPPMON	Stops the PDSMAN Partitioned Resource Monitoring subtask
NEWPMON	Bounces the PDSMAN Partitioned Resource Monitoring subtask

F PDSMAN,*command* ()

- The following are the PDSMAN modify commands:

Command	Description
PMONSTATUS	Displays the current status of the PDSMPMON subtask
PMONINFO	Displays status and other information for the PDSMPMON subtask
PMONDETAIL	Displays detailed information about the PDSMPMON subtask
PMONPERFORM	Displays performance information about the PDSMPMON subtask
STARTDB	Starts the PDSMAN Database Control subtask and the Database address space
STOPDB	Stops the PDSMAN Database Control subtask and the Database address space
NEWDB	Bounces the PDSMAN Database Control subtask and the Database address space
DBSTATUS	Displays the current status of the PDSMDB subtask and the Database address space
DBINFO	Displays status and other information for the PDSMDB subtask and the Database address space
DBDETAIL	Displays detailed information about the PDSMDB subtask and the Database address space
DBPERFORM	Displays performance information about the PDSMDB subtask and the Database address space

F PDSMAN,*command* ()

- The following are the PDSMAN modify commands:

Command	Description
DBCLOSE	Closes and de-allocates the database file
DBOPEN	Allocates and opens the database file
DBQUIESCE	Places database access or update requests into a wait state
DBRESTART	Releases queued database access or update requests for processing (negates the affect of the DBQUIESCE command)
CLEARDBM	Resets database server-related control block pointers. • Note: This command is for use only at the direction of Technical Support
STARTQMDB	Starts the PDSMAN Database Queue Manager subtask
STOPQMDB	Stops the PDSMAN Database Queue Manager subtask
NEWQMDB	Bounces the PDSMAN Database Queue Manager subtask
QMDBSTATUS	Displays the current status of the PDSMQMDB subtask
QMDBINFO	Displays status and other information for the PDSMQMDB subtask
QMDBDETAIL	Displays detailed information about the PDSMQMDB subtask
QMDBPERFALL	Displays current and recorded performance information about the PDSMQMDB subtask

F PDSMAN,*command* ()

- The following are the PDSMAN modify commands:

Command	Description
STARTHCHK	Starts PDSMAN Health Check services
STOPHCHK	Stops PDSMAN Health Check services
NEWHCHK	Bounces PDSMAN Health Check services
HCHKSTATUS	Displays the current status of PDSMAN Health Check services
DYNOFF	Suspends Dynamic BLDL processing for the specified library
DYNON	Resumes Dynamic BLDL processing for the specified library
DYNRESET	Resets the Dynamic BLDL table for the specified library or for all tables
RESET	Resets the Dynamic BLDL table for the specified library or for all tables

Links

PDSMAN

Links
PDSMAN – EZYEDIT – Introduction
PDSMAN – EZYEDIT – Selection Panel
PDSMAN – EZYEDIT – Dataset List
PDSMAN – EZYEDIT – Member List
PDSMAN – EZYEDIT – SPACE
PDSMAN – EZYEDIT – TSO Command Shell
PDSMAN – Introduction
PDSMAN – Member Versions
PDSMAN – FASTCOPY
PDSMAN – PDSEASY
PDSMAN – PDSM18
PDSMAN – PDSMURPT

Other

Links
ISPF – An Experienced User Shares His Secrets
ISPF – Workstation Agent (WSA)
OPSMVS – OPSLOG Overview
REXX and ISPF – Troubleshooting
SDSF – Beyond the Basics

Conclusion

Final Comments

- PDSMAN provides a wide range of solutions to PDS and PDSE issues users face on a daily basis
  - Saves time and increases productivity
- Every PDSMAN/EZYEDIT release introduces new features and commands which are easy to understand, implement and use
- The tutorials and manuals are exceptional
  - Easy to read and understand
  - Suggest downloading the manuals periodically to obtain new updates

Thank You!

- A special thank you to the CA PDSMAN team for their technical input and assistance

- Suggest checking out the CA PDSMAN community web site at:

  CA – Mainframe Community – PDSMAN

  Or

  CA – Mainframe Community

  Select 'CA PDSMAN PDS Library Management' from the 'Products Covered' section located on the left hand side of the web page

Any questions or comments please contact

Bruce Koss

Wells Fargo Mainframe Operating Systems Support

Bruce.Koss@WellsFargo.com

(704) 600-8416

#SHAREorg

SHARE is an independent volunteer-run information technology association that provides **education**, professional **networking** and industry **influence**.