

PDSE Nuts and Bolts

Speaker: Thomas Reed /IBM Corporation

SHARE Seattle 2015

Session: 16956



#SHAREorg



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



Disclaimer

This presentation is best understood by considering that the author cannot discuss confidential IBM information. The information contained herein should be considered conceptually accurate but not implementation details.

It is easiest to qualify most statements by adding ‘mostly’, ‘somewhat’, ‘pretty much’, and ‘by and large’ to the end. The more vague a statement by the author sounds, the more qualifiers should be added.

Agenda

- The PDSE Data Set Structure
 - Version 1 Data Sets
 - Version 2 Data Sets
- The PDSE Address Space
 - FAMS
 - Index Management (IMF)
 - Buffer Management (BMF)
 - HIPERSPACE Caching
 - Serialization (CLM & SLS)



What is a PDSE?


- PDSE: Partitioned Data Set Extended
- A PDSE is a homogenous collection of directory and data pages
- PDSE server consists of one or two address spaces (SMSPDSE and SMSPDSE1)
- The SMSPDSE(1) address spaces serve client access requests for PDSE data sets
- Under the hood SMSPDSE(1) also manages PDSE serialization and buffering

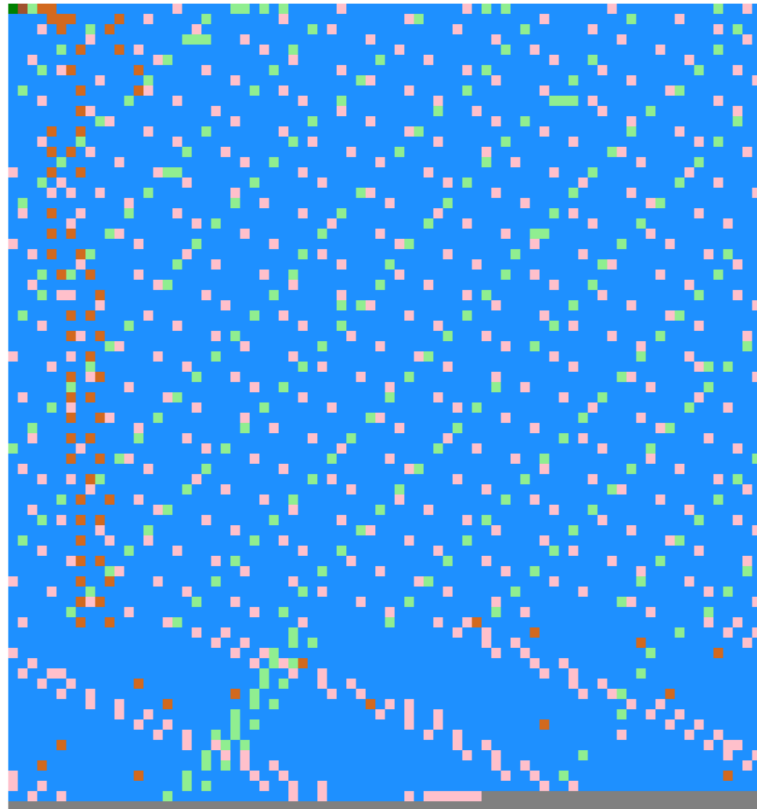
What is a PDSE?

- PDSE: Partitioned Data Set Extended
- A PDSE is a homogenous collection of directory and data pages
- PDSE server consists of one or two address spaces (SMSPDSE and SMSPDSE1)
- The SMSPDSE(1) address spaces serve client access requests for PDSE data sets
- Under the hood SMSPDSE(1) also manages PDSE serialization and buffering

The PDSE Dataset Structure

PDSE Dataset Structure : What Does a PDSE Look Like?

-  VDF
-  ND
-  NOTFMT
-  BMF
-  MEMBER
-  Free
-  LOST
-  AD



PDSE Dataset Structure :

Page Concepts

- On DASD:
 - PDSE's are a homogenous collection of 4K pages
- **A page is a page**, no matter what type of data it contains
 - Everything occurs at the page level, I/O, Buffering, and Index Trees
- Pages contain records
 - Can contain any sequential data
 - This includes index records
 - Can be fixed or variable length

PDSE Dataset Structure :

PDSE Directories

- Analogous to the classic PDS directory
 - Gets you from a member name to your data
 - Gets you to your data faster than a linear search
- Otherwise the PDSE directory is almost entirely different from the PDS directory

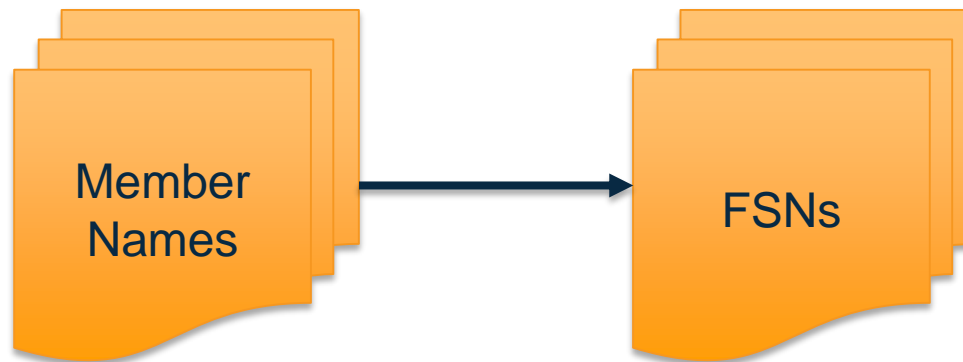
PDSE Dataset Structure :

PDSE Directories

- The PDSE directory is actually several b-tree style directories:
 - The Name Directory (ND)
 - The Attribute Directory (AD)
- The Version 2 format adds a third:
 - The Generation Directory (GD)
- Why so complicated?!?
 - Space efficiency
 - Search efficiency
 - Serialization and integrity

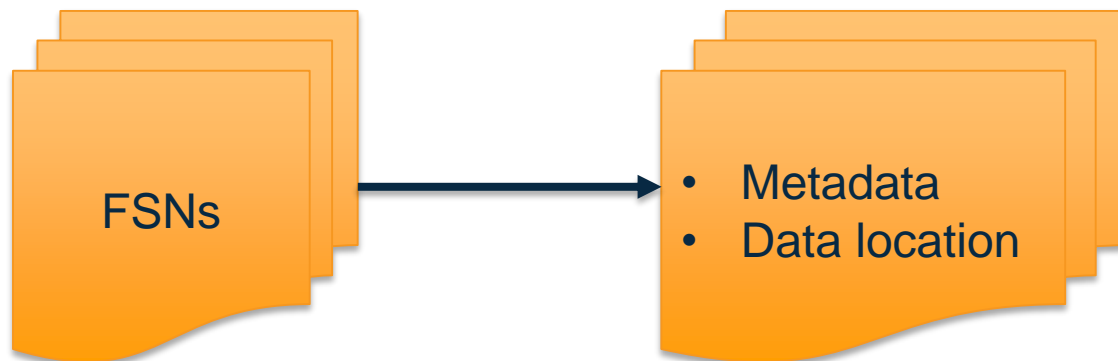
PDSE Dataset Structure : The Name Directory

- Indexed on the actual member names
- Relates a member name to the File Serial Number (FSN) that is associated with it
 - The FSN is the real identifier for each member
 - In PDSE-land names aren't particularly important



PDSE Dataset Structure : The Attribute Directory

- Contains two important chunks of information:
 - The member's metadata
 - Creation time
 - Last change time
 - Size
 - Etc.
 - The location of (the location of) the actual member pages
- Indexed by the File Serial Number (FSN)
 - A relatively arbitrary value
 - Related to the member name by the Name Directory



PDSE Dataset Structure : Directory Leftovers

- The Virtual Storage Group Descriptor Frame (VDF) Page
 - A ‘helper’ page
 - Stores metadata for managing the guts of the PDSE
 - Helps to keep track of where all the other pages are in the PDSE

- The Buffer Management Facility (BMF) Page
 - Another ‘helper’ page
 - Keeps track of usable pages within the dataset

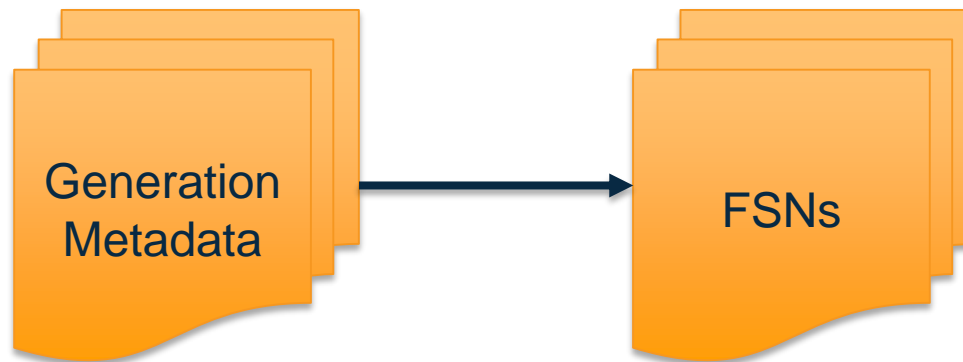
PDSE Dataset Structure : Version 2 Format Changes

- The Version 2 format was introduced with z/OS 2.1
- V2 is a streamlining of the V1 format
- The formats are largely indistinguishable

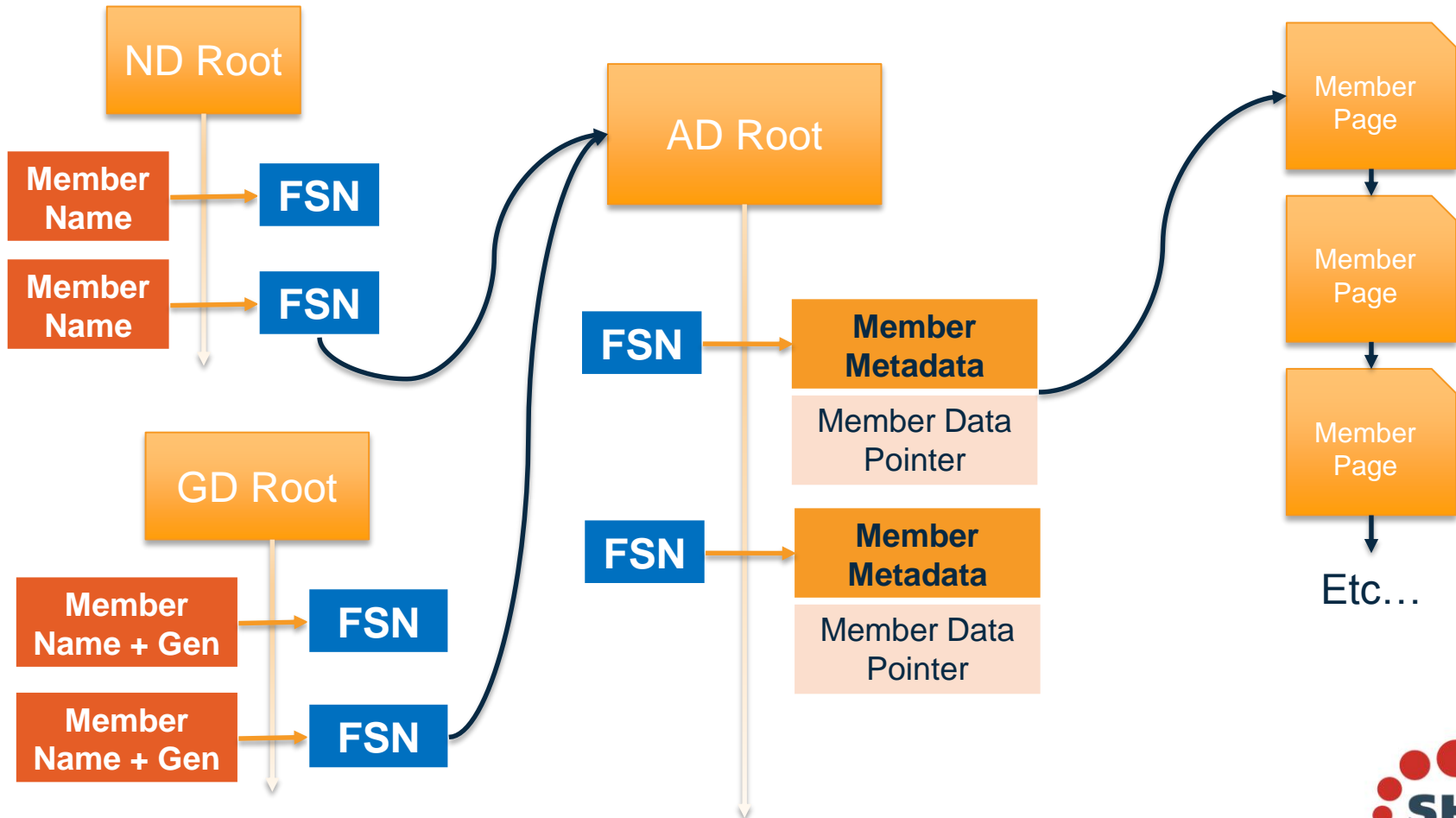
- The fundamentals of the index remain the same
 - Removed layers of indirection
 - Simplified the process of resolving page locations when traversing the directories
- Consolidated metadata
 - Data set statistics are moved into the tree roots
 - Simplifies the process of finding/updating the metadata

PDSE Dataset Structure : The Generation Directory

- Only exists in V2 PDSEs with member generations
- Keeps track of all generations for each member
- Created only as needed to minimize index size
- Almost like an alternate index for the Name Directory



PDSE Dataset Structure : Putting the Directory Together to Find Your Data!



PDSE Dataset Structure : Opening a PDSE

- The Virtual Storage Group Token(VSGT)
 - How PDSEs are uniquely identified
 - Data set names are not important to PDSE
 - Internally this is how PDSE refers to data sets
- The VSGT Format:

01-VOLSER-TTTTRR

TTR of the
Format-1 DSCB

PDSE Dataset Structure : Opening a PDSE

- The first 5 pages:
 - A newly allocated PDSE will always start with 5 pages
 - These pages' locations are fixed
 - They are loaded into storage on every first open of a PDSE
 - They allow us to bootstrap ourselves into the directories
 - After we have these 5 pages we can dynamically get to anywhere else in the data set



Simple!
(If only it were single threaded...)

The PDSE Address Space (x2)

The PDSE Address Space: Introduction

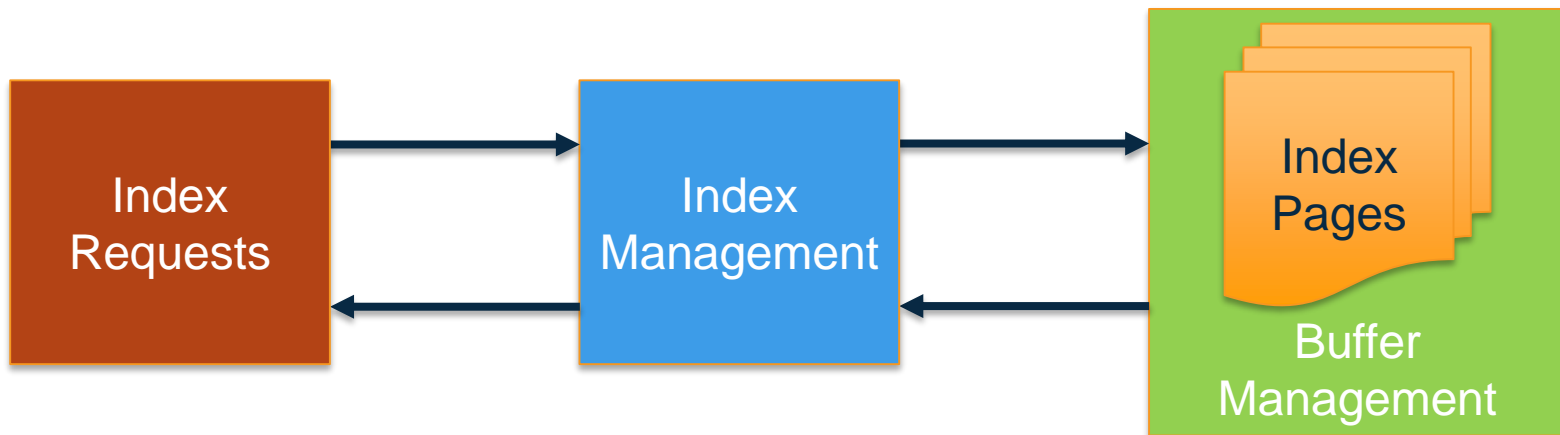
- The original PDSE address space
 - SMSPDSE, ASID 8
 - Generally handles global requests, typically loading from LNKLST
 - Will handle all PDSE requests if SMSPDSE1 is not present
 - Non-restartable
- The restartable PDSE address space
 - SMSPDSE1, Initially ASID 9
 - Handles non-global requests
 - Can be restarted to alleviate conditions that might otherwise require an IPL
- Both address spaces are essentially identical
 - Address the same functions, just for different types of connections
 - Peer address spaces/servers

The PDSE Address Space: FAMS

- File and Attribute Management Services (FAMS)
 - Set of interfaces for managing PDSE data sets
 - Manages data set and member attributes
 - Available through the Advanced Customization Guide
 - Implemented by many utilities i.e. DSS, IEBCOPY, ISPF, etc.
 - Provides the copy and conversion services for PDSEs
 - PDSE load and unload processing
 - IEBCOPY
 - DSS Logical Dump
 - Load Module to Program Object conversion

The PDSE Address Space: IMF

- Index Management Facility (IMF)
 - The middle-man of PDSE
 - Recall that the PDSE directories are b-trees
 - IMF provides the functions to manage and traverse those trees
 - Operates on index pages in storage
 - Sits between index requests and the in storage page management

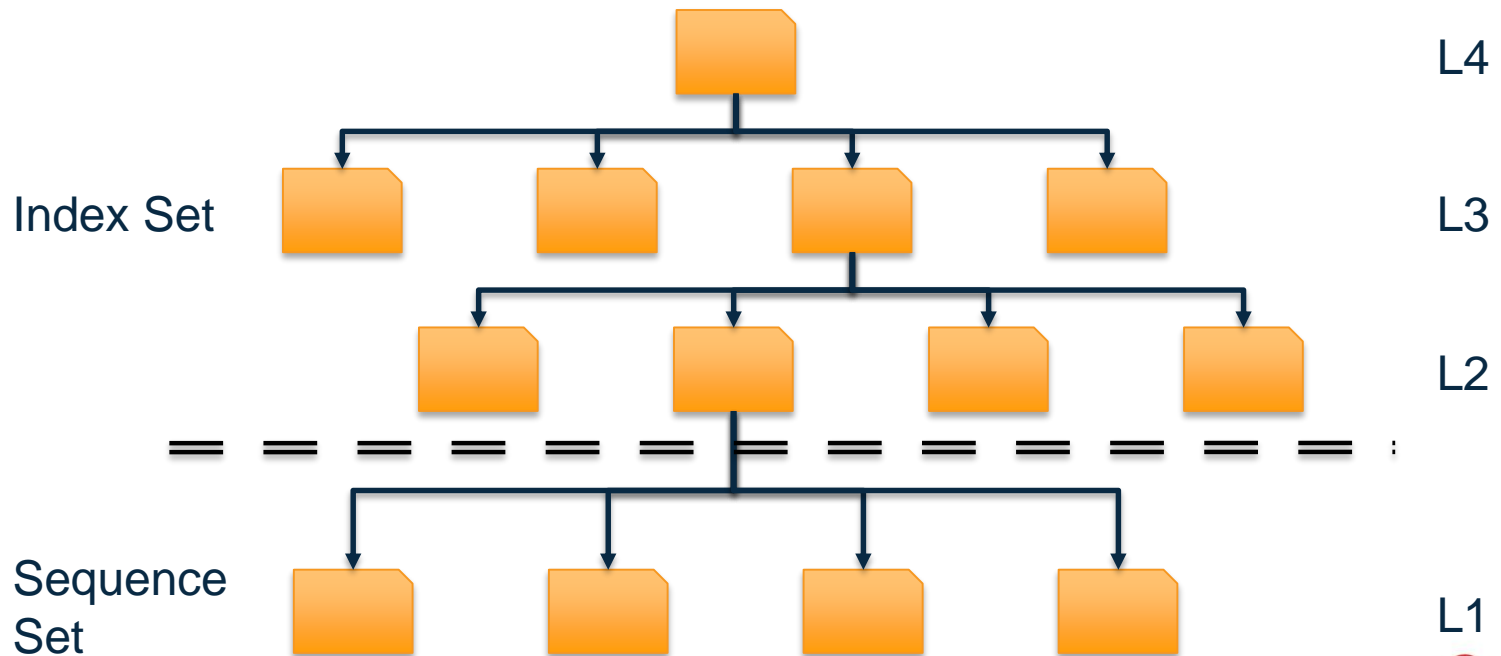


The PDSE Address Space: IMF

- PDSE Index Concepts
 - Keys:
 - A combination of one or more values of defined length
 - The attribute for which the index will be searched
 - Data:
 - Can contain actual data
 - Can contain a pointer to the actual data
 - Can contain a pointer to another index page
 - The PDSE index relates one key field to one data field
 - Together this relationship is referred to as an index record
 - Index Record: [KeyValue:DataValue]
 - Index Pages contain Index Records

The PDSE Address Space: IMF

- PDSE Index Concepts
 - Index pages are organized into balanced b-tree structures
 - Only the lowest level of the index contains actual data
 - All higher levels contain pointers into lower levels of the index



The PDSE Address Space: IMF

- IMF Function
 - IMF processes and maintains the index
 - Searches the index on a provided key
 - Adds/Removes/Updates index records as needed
 - Creates and Destroys index pages as needed
 - IMF and 2.1
 - IMF was rewritten at 2.1
 - Moved most index operations to inline macros (decreased path length)
 - Improved index record management algorithms
 - IMF index **updates are atomic**

The PDSE Address Space: BMF

- Buffer Manager Facility
 - Manages PDSE page I/O
 - Handles both reading and writing of all types of pages
 - Allows for both synchronous and asynchronous I/O
 - Records SMF records (type 42)
 - “Black Box” system
 - Write requests take a page and return status feedback
 - Read requests take a buffer and return a page in that buffer as well as status feedback
 - Implements a lookaside for directory pages
 - Improves read performance
 - Deferred writes
 - ALWAYS active

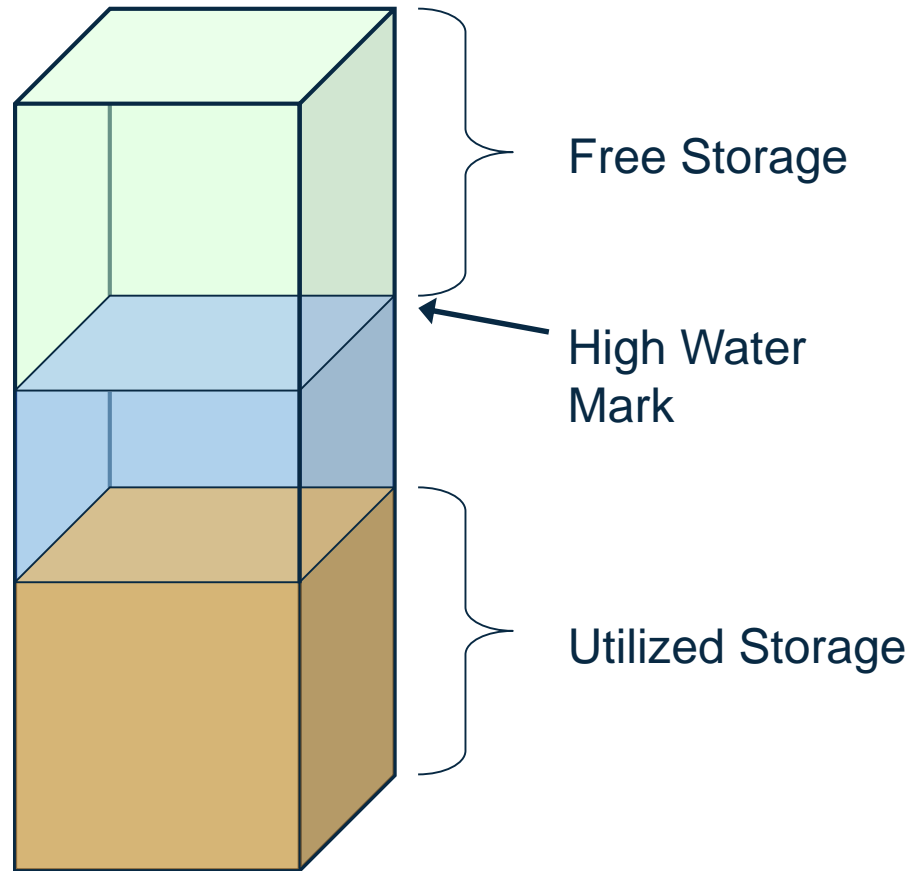
The PDSE Address Space: BMF

- The BMF Lookaside (cache)
 - AKA the PDSE directory cache
 - PDSE in-core cache
 - BMF cache
- Caching on Read requests
 - When possible, Reads for directory pages will be fulfilled through the cache
 - When a page is not resident in cache, BMF will automatically read the page from DASD and place it into the cache
 - All pages in the cache are guaranteed to be the latest copy
- Caching on Write requests
 - Writes occur to the cache first
 - When multiple directory pages are written they are held in cache until the caller requests they be committed

The PDSE Address Space: BMF

- The BMF lookaside continued...
 - The bulk of PDSE storage usage
 - Works on a high water mark system
 - Allocates as much storage as needed to fulfill concurrent caching requests
 - Pages are marked reusable when:
 - Invalidated due to update/replacement
 - Invalidated due to last close
 - Invalidated due to the LRU aging the page out

The PDSE Address Space: BMF



The PDSE Address Space: BMF

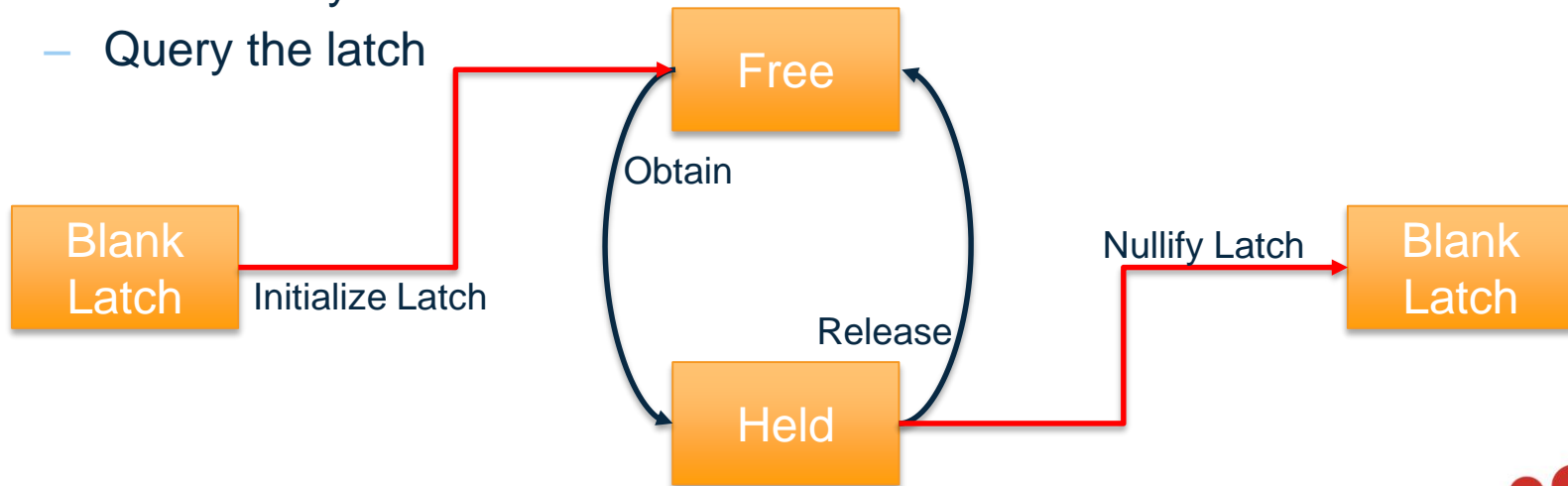
- HIPERSPACE caching
 - Just like the Directory cache but for member pages
 - Does not cache every member page
 - Only for data sets with the “Must Cache” flag set
 - OR told to cache by LLA
 - Uses the same LRU task as BMF
 - LRU settings are shared
 - LRU requires CPU cycles
 - Unlike the directory cache the HSP_SIZE parameter gives a hard size limit
 - Not currently enabled by default
 - **Hiperspace Caching Overview:**
 - <http://www.ibm.com/support/docview.wss?uid=isg3T1022058>

The PDSE Address Space: SLS

- Suspend Latching Services (SLS)
 - What is a latch anyways?
 - A double/quadword field in storage that is defined as a latch by the programmer
 - The latch address is also the latch's identifier
 - Generally latches are referred to by their offset into the control block that they reside in.
 - E.g. IGWHL1B+10 is the HL1B main latch
 - The latch's contents represent the state of the latch
 - Latches are EXCLUSIVE (AKA a mutex)
 - How are latches used?
 - Latches serialize access to PDSE control blocks
 - Latching is local to each CPC
 - The SLS component provides utility services to manage latches

The PDSE Address Space: SLS

- SLS supports the following latching operations:
 - Initialize the latch
 - Prepare the latch for use, place it in the FREE state
 - Obtain the latch
 - Release the latch
 - Nullify the latch
 - Destroy the latch
 - Query the latch



The PDSE Address Space: SLS

- What do we do when a latch is held but another task needs the latch?
 - Callers to latch obtain can choose to SUSPEND if a latch is held or fail.
 - Obtain Conditional (Can fail on a held latch)
 - Obtain Unconditional (Will wait on the latch)
 - Callers that SUSPEND will be placed in a waiter queue.
 - When a waiter is granted the latch, they are RESUME'd.

The PDSE Address Space: SLS

- Who uses latches anyway?
 - All PDSE internal components
 - Virtually every control block in the PDSE address space
 - Either has it's own latch
 - Or is chained to a root block with a latch
 - RLS latching is extremely closely related

- How are latches used?
 - Serialize data in a multi-threaded environment
 - Chaining dependency for PDSE control structures
 - Latches are generally held for milliseconds at a time
 - NOT to serialize PDSE data sets

The PDSE Address Space: CLM

- Common Lock Manager (CLM)
 - CLM is the mediator of locking requests
 - Client tasks initiate the locking request process
 - Begins with a connection for INPUT or OUTPUT
 - CLM obtains the locking state needed to complete the client request

Remember:

- Locks and latches are separate entities
 - Locks serialize PDSE data sets
 - Latches serialize PDSE control block structures
- Locks have a SYSPLEX scope

The PDSE Address Space: CLM

- What is a lock?
 - Not just an enqueue
 - A composite data structure
 - Resides in storage
 - Represents a serializable data set resource

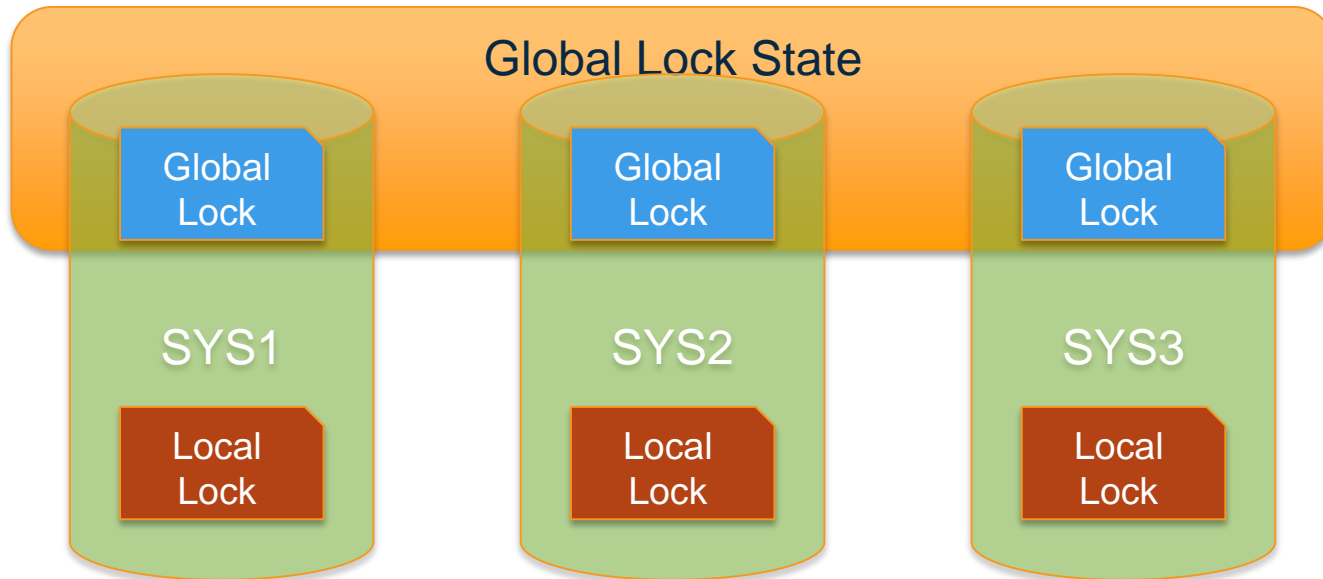
The PDSE Address Space: CLM

- A PDSE dataset has multiple serializable resources
 - Data Set locks
 - Local lock for serialization within a single system
 - Global lock for SYSPLEX-wide serialization
 - Structural locks
 - Directory lock
 - Format-write lock for moving/extending/formatting the data set
- PDSE uses GRS enqueues as well
 - On allocation
 - SYSDSN
 - On open
 - SYSZIGW0
 - Sharing protocol negotiation
 - SYSZIGW1

The PDSE Address Space: CLM

- Lock function depends on what sharing mode PDSE is using
 - NORMAL mode sharing
 - Locks act much like a regular enqueue
 - EXTENDED mode sharing
 - Locks become much more complicated
- Locks are held by SYSTEMS not by tasks
 - The LOCAL lock state determines that system's level of access to a PDSE
 - The LOCAL lock state must be compatible with the GLOBAL lock state
 - A system must have obtained both the LOCAL and GLOBAL lock states that it needs before it can use the PDSE
 - This relationship forms the “Hierarchical” portion of PDSE locking

The PDSE Address Space: CLM



The PDSE Address Space: CLM

- Locking in NORMAL mode sharing
 - Locks behave much like enqueues
 - Only has 2 lock states held for the SYSTEM that the open is occurring on.
 - Only Readers (OR): System may have the PDSE open for any number of INPUT operations but no OUTPUT operations. If the PDSE is already open in EX state on ANOTHER system then the system will fail to obtain the OR lock. OR lock may be promoted to an EX lock.
 - Exclusive (EX): The system may perform any INPUT or OUTPUT on the PDSE. The holding system is the only system that may access the PDSE. The EX lock is obtained on open for OUTPUT.
- Locking is simple and no need for messaging
- Only allows for sharing at the data set level between systems

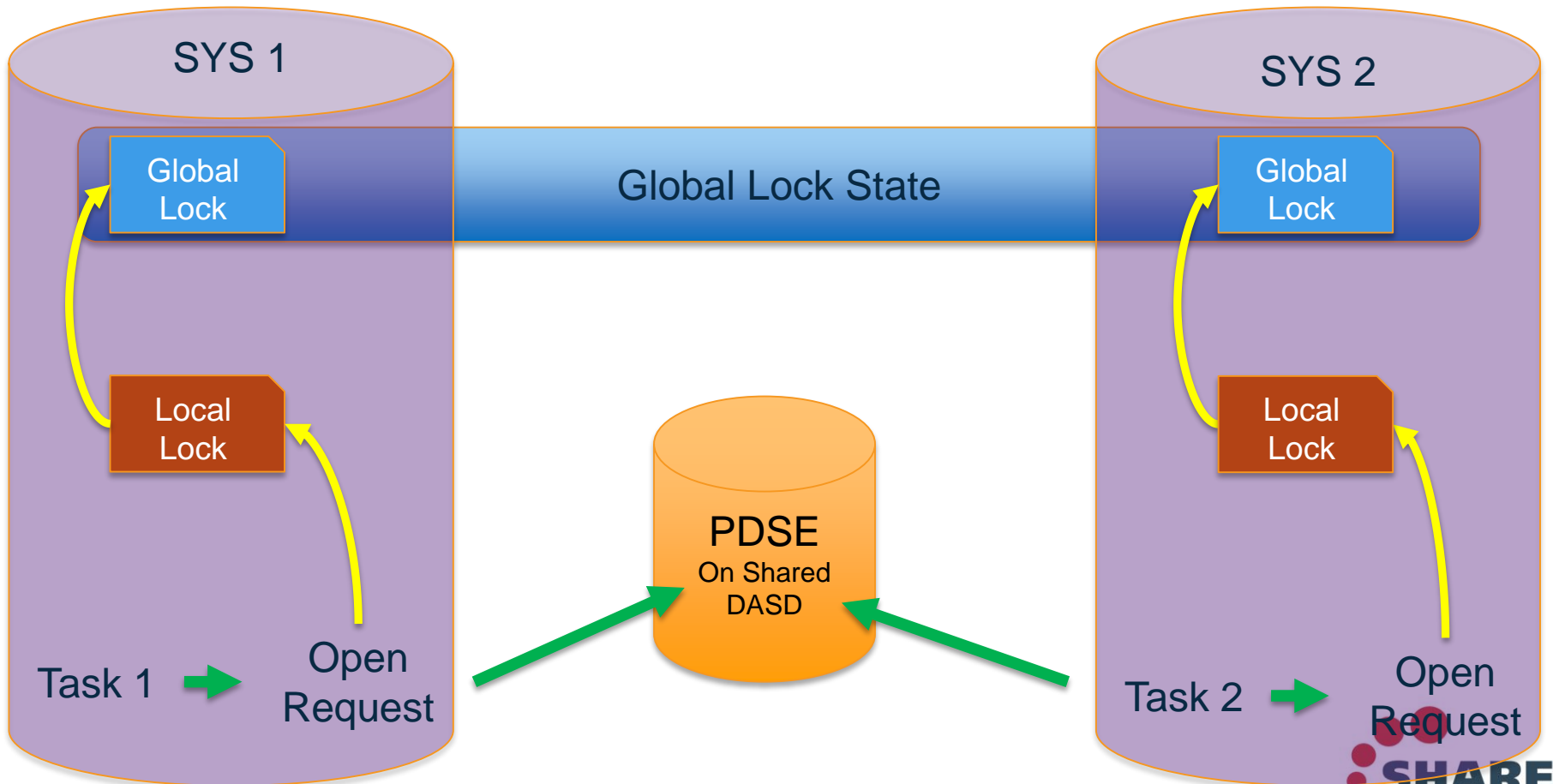
The PDSE Address Space: CLM

Normal Mode Sharing Between Systems

Second Job opens for:	First Job has the PDSE open for:		
	Input	Output	Update
Input	Success	Open Failure	Open Failure
Output	Open Failure	Open Failure	Open Failure
Update	Open Failure	Open Failure	Open Failure

Open Failure = IEC143I ABEND 213-70

The PDSE Address Space: CLM



The PDSE Address Space: CLM

- Locking in EXTENDED mode sharing
 - Supports member level sharing between systems
 - Accommodates locking requests that would otherwise fail with NORMAL mode
 - How do we do this?
 - More lock states for greater granularity
 - Negotiable lock states to accommodate multiple systems' access requests
 - Requires XCF for communication
 - Passes locking information between PDSE address spaces
 - Communicates
 - Lock holders
 - Lock waiters
 - Lock contention

The PDSE Address Space: CLM

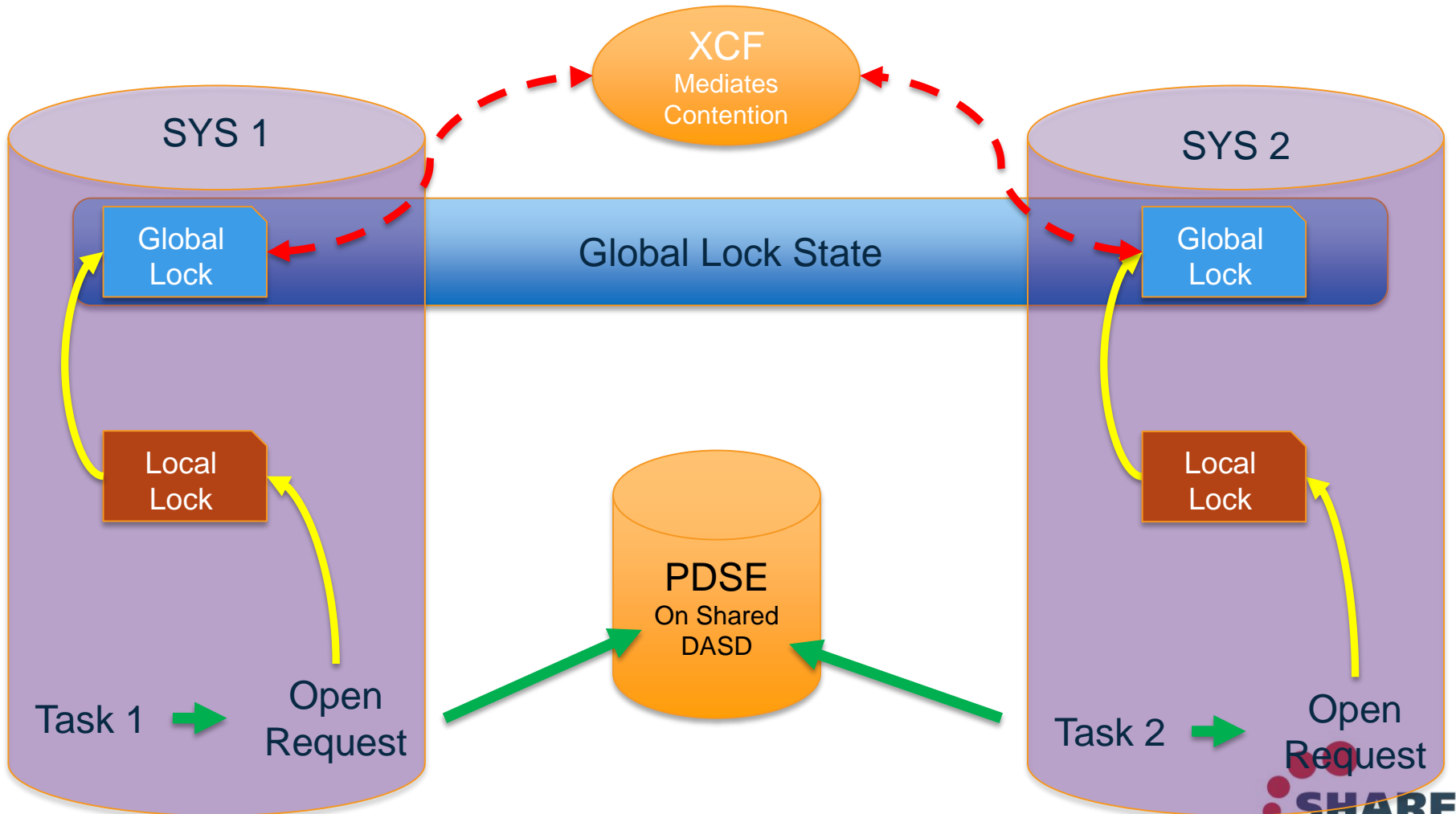
Extended Mode Sharing Between Systems

Second Job opens for:	First Job has the PDSE open for:		
	Input	Output	Update*
Input	Success	Success	Success
Output	Success	Success	Success
Update	Open Failure	Open Failure	Open Failure

Open Failure = IEC143I ABEND 213-70

* = When not positioned to a member

The PDSE Address Space: CLM



Questions? Comments?



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



Please Fill Out the Survey!



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