PDSE Member Generations: Implementation and ISPF Exploitation

Speakers: Thomas Reed and Sam Reynolds /IBM Corporation
SHARE Orlando 2015
Session: 17831
Agenda

- PDSE Member Generations Basics
- Working with Member Generations
- Member Generations Macro Support
- ISPF Support
What is a PDSE?

• PDSE: Partitioned Data Set Extended
• A PDSE is a collection of directory and data pages
• At V2R1 there are 2 data set formats V1 and V2 PDSEs
• 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
PDSE Dataset Structure: What Does a PDSE Look Like?
PDSE Versions

• The version 1 format is the historic PDSE format
• The version 2 format is a revision of the PDSE format
  – Brings better performance and efficiency
  – Reduces CPU and Storage utilization
  – **Supports PDSE member generations**

• Version 2 data sets use the same serialization and buffering subsystems as version 1
• The IMF/BMF performance enhancements at V2R1 apply to BOTH V1 and V2 datasets
PDSE Version 2: Member Generations

- What does it do?
  - PDSE Data sets can now retain multiple generations of members
  - Applies to BOTH Data Members and Program Objects
  - Retains generations up to the data set/system limit

- Implemented via DFSMS APAR OA42358
  - ISPF Support via APARs OA42247 and OA42248
PDSE Version 2: Member Generations

Terminology

• Generation (GEN)
  – A prior copy of a member

• Primary Generation/Member
  – The current member
  – Absolute and Relative 0

• MAXGENS_LIMIT
  – IGDSMSxx Parameter

• MAXGENS
  – Set at allocation time
  – MAXGENS <= MAXGENS_LIMIT
PDSE Member Generations: The Basics

- FIFO (First In, First Out) structure
  - Oldest generation is permanently deleted if it’s over the generation limit
  - Old generations generally behave just like primary members
  - Aliases are retained for previous generations and can be recovered*

* When STOW RECOVERG is used

Complete your session evaluations online at www.SHARE.org/Orlando-Eval
PDSE Member Generations: The Basics

• Generations are uniquely numbered
  – They can be referenced either by their **Absolute** or **Relative** generation
  – The Primary Member is always 0, both relative and absolute
  – Greatest number indicates the newest generation

• Generation Numbering
  – Absolute: GEN(n), GEN(n-1), GEN(n-2)…. 
  – Relative: GEN(-1), GEN(-2),…..,GEN(-MAXGENS)
    • n being the nth generation created
PDSE Member Generations: The Basics

- Example: MAXGENS = 4 after 11 generations

<table>
<thead>
<tr>
<th>ABS</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRI</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>Gen</td>
<td>Gen</td>
<td>Gen</td>
<td>Gen</td>
</tr>
<tr>
<td>REL</td>
<td>0</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-4</td>
<td>-3</td>
<td>-2</td>
<td>-1</td>
</tr>
</tbody>
</table>

- Note that the newest generation ALWAYS has the greatest value
PDSE Member Generations: The Basics

Parameters

- **MAXGENS_LIMIT** can be set dynamically via SET SMS=xx
  - Cannot be set dynamically with SETSMS
- **MAXGENS_LIMIT** upper limit is set at 2 billion
  - Limit is mostly theoretical.
  - In total, 2 billion generations can have ever existed in a V2 PDSE
- PDSE honors the MAXGENS on a PDSE regardless of the MAXGENS_LIMIT on that system
  - MAXGENS is set at allocation time and is stored in the PDSE
  - MAXGENS cannot currently be modified
PDSE Member Generations: The Basics

Usage Considerations

• Allow extra space for each generation
• Each generation retains the entire member
  – Generation creation is atomic
  – Mirrors the “normal” PDSE member replace process
  – Simply swaps pointers rather than doing IO for the whole member
• Deleting generations can result in pending deletes, just like primary members
Copy Basics

• DSS
  – DSS Copy will retain generations with OA43729 or Concurrent Copy
  – Logical or Physical DUMP and RESTORE retains generations
• IEBCOPY and IDCAMS REPRO
  – Does NOT retain generations
  – ONLY copies primary members
PDSE Member Generations: Working with Generations

Enabling Member Generations

1. Ensure that the requisite APARs are applied
2. MAXGENS_LIMIT needs to be set >0 in your IGDSMSxx
3. Allocate a V2 PDSE dataset with greater than 0 generations (must be <= MAXGENS_LIMIT)
PDSE Member Generations: Coexistence

• Down level systems will tolerate V2 PDSE’s with Generations and be able to open for INPUT of OUTPUT
• Down level systems will not be able to create or manipulate generations
• Down level systems are essentially unaware of generations while maintaining compatibility
• DFSMSdss/IEBCOPY support is identical to 2.1
Allocating a PDSE with Generations Enabled via JCL!

```
//ALLOC    EXEC PGM=IEFBR14
//PDSE2    DD DSN=TREED.PDSE.GENS,
//          DSNTYPE=(LIBRARY,2),MAXGENS=10,
//          RECFM=FB,LRECL=80,
//          UNIT=SYSALLDA,SPACE=(CYL,(1,1,1)),
//          DISP=(,CATLG,DELETE)
```

- Note that LIBRARY,2 specifies a V2 data set
- MAXGENS must be <= the system MAXGENS_LIMIT
PDSE Member Generations: Working with Generations

Creating a Generation

• 2 requirements
  – (LIBRARY,2)
  – MAXGENS > 0

• New generations are automatically created on replace or delete of a member
• Update in place will not create a new generation
• Generation creation is atomic
Reading Old Generations

- FIND macro will allow programs to connect to old generations
- Conventional READ and CHECK macros still apply
- Old generations cannot be accessed via JCL or dynamic allocation
PDSE Member Generations: Working with Generations

Deleting Old Generations

- Each generation must be deleted separately
- Deleted generations can be replaced by using STOW RG
- ISPF member delete will delete all generations
PDSE Member Generations: Working with Generations

Recovering Old Generations

• Read an old generation and then write it to either the same or a different member name
  – The old generation will become the current generation
  – Note: This method will not restore aliases

• Use the RECOVERG option for the STOW macro
  – The old generation becomes the current generation of the member of the same name
  – Note: Aliases ARE recovered by this method
PDSE Member Generations: Working with Generations

Backup Considerations

- **HSM Backup**
  - Uses DSS under the covers to move the PDSE
  - Retains all generations

- **What is the difference between IEBCOPY and DSS Copy for PDSE?**
  - DSS does a page copy of the entire dataset
    - Does not care about the contents of the PDSE
  - IEBCOPY process members individually from the index
    - Can merge sets of members
    - Merging of generations is beyond the scope of IEBCOPY
PDSE Member Generations: DESERV Macros

FUNC=GET_G (AKA Get Generation)

- Returns information for the selected generation
- Returns the same information as GET plus the relative and absolute generation numbers
- A dummy entry is returned if the selected generation does not exist
- Does not support CONNECT

Complete your session evaluations online at www.SHARE.org/Orlando-Eval
PDSE Member Generations:
DESERV Macros

FUNC=GET_G

,AREA=(buffer_area, buffer_area_size)
,DCB=data_control_block
,NAME_LIST=(generationname,1)
,[,MF={((E,parmlist_name[,NOCHECK|COMPLETE]])|S}]
,[,RETCODE=return_code]
,[,RSNCODE=reason_code]
PDSE Member Generations: DESERV Macros

FUNC=GET_ALL_G (AKA Get All Generations)

- Returns information for the selected generation for all members
- Returns the same information as GET_ALL plus the relative and absolute generation numbers
- A dummy entry is returned if the selected generation does not exist for a member
- Does not support all the same options as GET_ALL
FUNC=GET_ALL_G

,AREA=(buffer_area, buffer_area_size)
,DCB=data_control_block
,NAME_LIST=(generationname,1)
[,MF=\{(E,parmlist_name[,NOCHECK|COMPLETE]]|S}\}]
[,RETCODE=return_code]
[,RSNCODE=reason_code]
PDSE Member Generations: STOW Macro

DG (Delete Generation)
- Deletes an existing generation
- Takes a member name and generation number
- Leaves a gap in the generation list
- If issued with a generation of 0, deletes the member without creating a generation
PDSE Member Generations: STOW Macro

RG (Replace Generation)

- Replaces an existing generation
- Adds a generation if replacing a gap in the generation list
PDSE Member Generations: STOW Macro

RECOVERG (Recover Generation)

- Recovers an existing generation
- Removes the selected generation from the generation list and makes it the primary member
- Creates a new generation in the replace process from the former primary member
ISPF and PDSE Member Generations
PDSE Member Generations: ISPF Support

Panels

• ISPF has generations support with APARs OA42247 and OA42248 on z/OS V2R1

• Enhanced member list option must be selected
PDSE Member Generations:
ISPF Support
Allocation

• Allocates like any other PDSE
• MAXGENS must be >0
• Be sure you’re using version 2!
PDSE Member Generations: ISPF Support

- Accessing generations through the member list

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Prompt</th>
<th>Size</th>
<th>Created</th>
<th>Changed</th>
<th>Treed</th>
</tr>
</thead>
<tbody>
<tr>
<td>e</td>
<td>AMEMBER</td>
<td>/</td>
<td>1</td>
<td>2014/07/30</td>
<td>2014/07/30 13:57:19</td>
<td>TREED</td>
</tr>
<tr>
<td></td>
<td>BMEMBER</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>End</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Don't forget the ‘/’

EDIT Entry Panel

Object Name: 'TREED.GENTST(AMEMBER)'

- No workstation connection
- Initial Macro...
- PDSE Generation...
- Line Command Table
- Profile Name...
- Format Name...
- Panel Name...

Options

- Confirm
- Cancel
- Move
- Replace
- EDIT Mixed Mode
- EDIT host file on
- Workstation

Data Encoding

- 1. ASCII
- 2. UTF-8

/ Preserve VB record length

F1=Help  F2=Split  F3=Exit  F7=Backward  F8=Forward
F9=Swap  F12=Cancel
PDSE Member Generations: ISPF Support

Restrictions

• ENQUEUEing on one generation applies to all generations of that member
  – This is not a PDSE serialization restriction
  – The native API’s allow for editing of multiple generations of the same member

• ISPF Options 1 and 2 do not support a GEN parameter

• ISPF 3.1 and 3.4 do support a GEN parameter
PDSE Member Generations: ISPF Support

Editing

- Editing the current member without specifying a generation (GEN 0) results in a new generation being created
- Editing prior generations does NOT result in a new generation
- Supports referencing generations by either absolute or relative generation number
- Deleting a member in ISPF deletes all generations
  - This is an ISPF implementation feature
  - TSO DELETE pdse(member) deletes only the primary
- Renaming a member deletes all generations except the base generation
- COPY only copies the base generation
PDSE Member Generations: ISPF Support

Editing Cont’d

• Generation creation behavior can be forced
  – SAVE NEWGEN – Creates a new generation (default action for base generation)
  – SAVE NOGEN – Does not create a new generation when saving base generation

• Edit will tell you which absolute generation you are working with

```
EDIT TREED.GENTST2(TST1) - 01.00
```

```
******** Top of Data ***********
==MSG== Warning: The UNDO command is not available until you change
     your edit profile using the command RECOVERY ON.
==MSG== CAUTION: Edit session has been invoked for generation 1
     High generation number is currently 2
```

Complete your session evaluations online at www.SHARE.org/Orlando-Eval
PDSE Member Generations: Example

- Original set of generations:
  
  **Generation 0**
  The cat in the hat comes back

  **Generation -1**
  The cat in the hat

  **Generation -2**
  The cat

- We edit generation -2, changing it to “My cat”
- SAVE NEWGEN
- Now the generations are:

  **Generation 0**
  My cat

  **Generation -1**
  The cat in the hat comes back

  **Generation -2**
  The cat in the hat
PDSE Member Generations: Example …

• We edit the base generation, changing it to “My cat in the hat”
• SAVE NOGEN
• Now the generations are:

<table>
<thead>
<tr>
<th>Generation 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>My cat in the hat</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Generation -1</th>
</tr>
</thead>
<tbody>
<tr>
<td>The cat in the hat comes back</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Generation -2</th>
</tr>
</thead>
<tbody>
<tr>
<td>The cat in the hat</td>
</tr>
</tbody>
</table>

• If we rename the file, all generations except for the base are deleted:

![Invalid GEN parameter message]
PDSE Member Generations: PDF Support

- PDF API Changes for Generations
  - DSINFO
    - ZDSDSNV = The version of the PDSE
    - ZDSNGEN = The number of generations specified (MAXGENS) on allocation of the PDSE
  - LMDLIST
    - ZDLDSNV = The version of the PDSE
    - ZDLNGEN = The number of generations specified (MAXGENS) on allocation of the PDSE
  - EDIT, VIEW, and BROWSE support the GEN parameter
    - GEN(n) = Either relative or absolute generation
  - EDIT SAVE supports NOGEN and NEWGEN
PDSE Member Generations: Future Enhancements to ISPF Support?

• We would like to improve the usability of the ISPF support for PDSE Member Generations in the future
• Currently two RFEs are open requesting enhancements to the support:
  – RFE 55041: ISPF SAVE NEWGEN add message indicating number of generations
  – RFE 55908: ISPF edit member generation message
• We would welcome additional RFEs in this area with your suggestions for improvements

Complete your session evaluations online at www.SHARE.org/Orlando-Eval
By now most of you should be familiar with the RFE (Request for Enhancements) process for submitting and voting on requirements against ISPF, z/OS, and many other IBM products.

- We are in the planning stage for the next release (V2R2+). So, now is the time to vote and/or let us know of any new requirements.

- Go to: https://www.ibm.com/developerworks/rfe/

- Select Brand = Servers and Systems Software, and Product = ISPF
PDSE Requirements Survey

• PDSE is looking for your feedback!

• Survey and item descriptions are in the handout section for this presentation.
Questions? Comments?

Complete your session evaluations online at www.SHARE.org/Orlando-Eval
Please Fill Out the Survey!

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