z/OS PARMLIB Successful Practices - User Experience

Thomas Conley
Pinnacle Consulting Group, Inc.
59 Applewood Drive
Rochester, NY 14612-3501
P: (585)720-0012  F: (585)723-3713
pinncons@rochester.rr.com
http://home.roadrunner.com/~pinncons/
Thursday, August 11, 2011
Session 9800
Agenda

- WAC PARMLIB Philosophy
- WAC PARMLIB Member Update Method
- Summary
- Finally…
WAC PARMLIB Philosophy

- WAC is "World According to Conley"
- Hopefully WAC is not too wack
- WAC PARMLIB philosophy makes ease of installation and backout paramount, all other considerations are secondary
- Keep operator interaction to absolute minimum
  - Minimize HMC and console input for install and backout
  - Do not allow prompt for PARMS at IPL
  - Initialization Message Suppression Indicator (IMSI) character
  - z/OS System Commands documents 7th position of load parm
  - IMSI 'M' overrides PROMPT for master cat and sysparsms
  - Load parm xxxx00M1
WAC PARMLIB Philosophy

- PARMLIB is not shared, and resides on primary res volume
- PARMLIB concatenation
  - SYS1.IBM.PARMLIB (members change only with maintenance)
  - SYS1.PARMLIB (customized PARMLIB on res volume)
- SYSy.IPLPARM(LOADxx) for new res

```plaintext
IODF  00 SYS1
SYSCAT ZTMCAT133CATALOG.MASTER.TEST
NUCLST 00
NUCLEUS 1
IEASYM  00
PARMLIB SYS1.IBM.PARMLIB
PARMLIB SYS1.PARMLIB
```

© Pinnacle Consulting Group, Inc., 2011. All rights reserved. Permission granted to SHARE to distribute for SHARE in Orlando 2011.
WAC PARMLIB Philosophy

- Active members are suffixed with "00"
- Backup members suffixed with "99"
- Installed with IPL, and backed out with IPL
- For install and backout, operator changes load address only
  - Minimizes input
  - No need for R 00, CLPA, SYSP=99, PROG=99, SVC=99, etc.
  - No need for LOADxx member pointing to backup PARMLIB
  - No need to modify LOAD parm for backup LOADxx member
- No need to copy or flip backup members in PARMLIB
- No need to use symbols, &SYSCLONE, etc.
WAC PARMLIB Member Update Method

- Standard PARMLIB update method has drawbacks
- Primary drawback is loss of update history
- WAC PARMLIB update method preserves update history
- What did they do, and when did they do it?
- Enables the guilty to be apprehended and punished
WAC PARMLIB Member Update Method

- **IBMUSER will modify EDGRMM00 and IKJTSO00**

<table>
<thead>
<tr>
<th>Name</th>
<th>Prompt</th>
<th>Size</th>
<th>Created</th>
<th>Changed</th>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>EDGRMM00</td>
<td></td>
<td>23</td>
<td>2011/06/22</td>
<td>19:46:21</td>
<td>TCONLEY</td>
</tr>
<tr>
<td>EDGRMM99</td>
<td></td>
<td>23</td>
<td>2011/06/20</td>
<td>15:31:00</td>
<td>IBMUSER</td>
</tr>
<tr>
<td>IKJTSO00</td>
<td></td>
<td>127</td>
<td>2011/06/22</td>
<td>19:50:41</td>
<td>TCONLEY</td>
</tr>
<tr>
<td>IKJTSO99</td>
<td></td>
<td>127</td>
<td>2011/06/20</td>
<td>15:30:00</td>
<td>TCONLEY</td>
</tr>
</tbody>
</table>

- **A common practice is to edit EDGRMM00 and to populate backup member EDGRM99 REPLACE C99999**
WAC PARMLIB Member Update Method

• After completing edit on EDGRMM00, stats for EDGRMM99 look like this:

<table>
<thead>
<tr>
<th>Name</th>
<th>Prompt</th>
<th>Size</th>
<th>Created</th>
<th>Changed</th>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>EDGRMM00</td>
<td></td>
<td>23</td>
<td>2011/06/22</td>
<td>2011/08/01 16:18:32</td>
<td>IBMUSER</td>
</tr>
<tr>
<td>EDGRMM99</td>
<td></td>
<td>23</td>
<td>2011/08/01</td>
<td>2011/08/01 16:16:49</td>
<td>IBMUSER</td>
</tr>
<tr>
<td>IKJTS000</td>
<td></td>
<td>127</td>
<td>2011/06/22</td>
<td>2011/06/22 19:50:41</td>
<td>TCONLEY</td>
</tr>
<tr>
<td>IKJTS099</td>
<td></td>
<td>127</td>
<td>2011/06/20</td>
<td>2011/06/20 15:30:00</td>
<td>TCONLEY</td>
</tr>
</tbody>
</table>

• Note how stats for EDGRMM99 (old EDGRMM00) are lost
• Created set to current date, Changed set to current date/time, and ID is IBMUSER instead of TCONLEY
WAC PARMLIB Member Update Method

- Let's edit IKJTSO00, but in a way that preserves history
- Edit IKJTSO00 and CREATE IKJTSO08 with C99999
- Cancel out of IKJTSO00 and edit IKJTSO08 for changes

<table>
<thead>
<tr>
<th>Name</th>
<th>Prompt</th>
<th>Size</th>
<th>Created</th>
<th>Changed</th>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>EDGRMM00</td>
<td></td>
<td>23</td>
<td>2011/06/22</td>
<td>2011/08/01 16:18:32</td>
<td>TCONLEY</td>
</tr>
<tr>
<td>EDGRMM99</td>
<td></td>
<td>23</td>
<td>2011/08/01</td>
<td>2011/08/01 16:16:49</td>
<td>IBMUSER</td>
</tr>
<tr>
<td>IKJTSO00</td>
<td></td>
<td>127</td>
<td>2011/06/22</td>
<td>2011/06/22 19:50:41</td>
<td>TCONLEY</td>
</tr>
<tr>
<td>IKJTSO08</td>
<td></td>
<td>128</td>
<td>2011/08/01</td>
<td>2011/08/01 16:28:20</td>
<td>IBMUSER</td>
</tr>
<tr>
<td>IKJTSO99</td>
<td></td>
<td>127</td>
<td>2011/06/20</td>
<td>2011/06/20 15:30:00</td>
<td>TCONLEY</td>
</tr>
</tbody>
</table>
WAC PARMLIB Member Update Method

• Delete IKJTSO99, rename IKJTSO00 to IKJTSO99, and rename IKJTSO08 to IKJTSO00

<table>
<thead>
<tr>
<th>Name</th>
<th>Prompt</th>
<th>Size</th>
<th>Created</th>
<th>Changed</th>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>EDGRMM00</td>
<td></td>
<td>23</td>
<td>2011/06/22</td>
<td>2011/08/01 16:18:32</td>
<td>IBMUSER</td>
</tr>
<tr>
<td>EDGRMM99</td>
<td></td>
<td>23</td>
<td>2011/08/01</td>
<td>2011/08/01 16:16:49</td>
<td>IBMUSER</td>
</tr>
<tr>
<td>IKJTS000</td>
<td></td>
<td>128</td>
<td>2011/08/01</td>
<td>2011/08/01 16:28:20</td>
<td>IBMUSER</td>
</tr>
<tr>
<td>IKJTS099</td>
<td></td>
<td>127</td>
<td>2011/06/22</td>
<td>2011/06/22 19:50:41</td>
<td>TCONLEY</td>
</tr>
</tbody>
</table>

• Note that IKJTSO99 retains original created date, original changed date and time, and original id from IKJTSO00
Summary

• Discussed speaker's PARMLIB philosophy
• Reviewed multiple different PARMLIB philosophies
• Discussed speaker's PARMLIB member update method
• Reviewed how to preserve PARMLIB update history
Finally…

- Let me know what you think and how you manage PARMLIB
- Please Email me with comments and/or questions
- pinncons@rochester.rr.com