z/OS V1R13

DFSMS PDSE: Verification tool (IEBPDSE)

Session objectives

- Learn about the new IEBPDSE utility to validate a PDSE data set

Overview

- Problem Statement / Need Addressed
  - Over the years there have been a number of customer requirements requesting the ability to verify the structural integrity of a PDSE. In some instances, customers back-up broken data sets as part of their routine operations, only to become aware of the problem months after the operation occurred, or multiple backups after the first one.
    - MR0125083823
- Solution
  - IEBPDSE. You can use the new utility to validate the directory of a PDSE before or after copying the data set to a new location.
- Benefit / Value
  - Allows you to know if the directory structure of a PDSE is valid.

Usage and invocation

- The PDSE validation utility may be invoked using job control statements. Like most utilities, IEBPDSE can be invoked from TSO if SYSLIB is allocated to a PDSE.
- EXEC statement invokes the PDSE validation utility using PGM=IEBPDSE. The PARM keyword may be specified.
- PARM=[DUMP|NODUMP] If the DUMP option is specified, the PDSE validation utility will issue an ABEND in the PDSE address space when an error has been found in the analysis of the PDSE.
- **Note:** The PDSE validation utility does not validate the data in the members.
- IEBPDSE does not require APF authorization.
- **Return Codes**
- 00 (X'00')   Successful completion.
- 04 (X'04')   Slightly damaged PDSE. The data set can be opened normally but has some form of corruption. Currently, the only instance of a “slightly” damaged PDSE is when the free space list marks free pages as used, wasting space that could normally be reclaimed. This does not prevent the user from opening the PDSE but the user should copy the PDSE to a new data set.
- 08 (X'08')   Corrupted PDSE.
- 12 (X'0C')   PDSE could not be opened.
- 16 (X'10')   Input data set not a PDSE.
- **The following JCL can be used to invoke the PDSE validation utility:**

```
//STEPCHK  EXEC PGM=IEBPDSE
//SYSPRINT DD SYSOUT=A
//SYSLIB   DD DSN=IBMUSER.SIMPLE.V2.PDSE,DISP=OLD
//STEPCHK2 EXEC PGM=IEBPDSE
//SYSLIB   DD DSN=IBMUSER.SIMPLE.V2.PDSE,DISP=OLD
//         DD DSN=IBMUSER.SIMPLE.V3.PDSE,DISP=OLD
//         DD DSN=SYS1.TCPIP.SEZALOAD,DISP=SHR
//STEPLINK EXEC PGM=IEBPDSE,PARM=’DUMP’
//SYSLIB   DD DSN=SYS1.TCPIP.SEZALOAD,DISP=SHR
```

- **Example 1: will validate IBMUSER.SIMPLE.V2.PDSE and send the results to SYSPRINT.**

```
//STEPCHK EXEC PGM=IEBPDSE
//SYSPRINT DD SYSOUT=A
//SYSLIB   DD DSN=IBMUSER.SIMPLE.V2.PDSE,DISP=OLD
```

- **Example 2: will validate IBMUSER.SIMPLE.V2.PDSE and IBMUSER.SIMPLE.V3.PDSE and send the results to the job log.**

```
//STEPCHK2 EXEC PGM=IEBPDSE
//SYSLIB   DD DSN=IBMUSER.SIMPLE.V2.PDSE,DISP=OLD
//         DD DSN=IBMUSER.SIMPLE.V3.PDSE,DISP=OLD
//         DD DSN=SYS1.TCPIP.SEZALOAD,DISP=SHR
```

Interactions and dependencies

- Software Dependencies
  - None
- Hardware Dependencies
  - None
- Exploiters
  - None

Migration and coexistence considerations

- None

Installation

- None

Session summary

- New PDSE utility is provided to validate the directory structure of a PDSE.

Appendix - References

- The Utility will be documented in:
  - *z/OS DFSMS Utilities,* SC26-7414-08