IBM Education Assistance for z/OS V2R3

IEFOPZ Support
Element/Component: BCP Allocation and Contents Supervisor

Material prepared for IEA and was current as of July 2017

Agenda

- Trademarks
- Session Objectives
- Overview
- Usage & Invocation
- Interactions & Dependencies
- Migration & Coexistence Considerations
- Installation
- Session Summary
- Appendix

Trademarks

- See url http://www.ibm.com/legal/copytrade.shtml for a list of trademarks.
- Additional Trademarks:
  - None

Session Objectives

- Understand the changes to provide z/OS IEFOPZ infrastructure
  - Allow customers to exploit the Automatic Binary Optimizer (ABO) for COBOL optimization

Overview

- Problem Statement / Need Addressed
  - ABO intends to optimize existing COBOL modules, taking advantage of machine's architecture
  - z/OS needs to provide infrastructure to help exploit the result of ABO without changing much JCL
- Solution
  - IEFOPZ processing via IEFOPZxx parmlib definition and support in Allocation and CSV to build concatenations of “NEW” ahead of “OLD” at appropriate times
- Benefit / Value
  - Improved performance without JCL updates

Usage & Invocation

- New IEFOPZxx parmlib member contains the IEFOPZ configuration, consisting of the following:
  - Maximum Architecture level (e.g., current machine's arch)
  - 1 or more OLD/NEW definitions
    - OLD data set
    - NEW data set (potentially a different one for each architecture level)
    - member(s) to be processed
  - 1 or more DDNAME / JOBNAME pairs
    - Allocation does special-case processing for JOBLIB, STEPLIB and any matching DDNAME/JOBNAME pairs
    - Has wildcard support
  - 1 or more OWNER / MinArch pairs identify the minimum architecture level that that OWNER supports
    - For ABO, that is 10 which corresponds to zEC12

Usage & Invocation (cont'd)

- IEFOPZ=(x1,...,xN) system parameter in IEASYSxx or IPL parms to initialize configuration
  - Last specification “wins” when multiple OLDs
- SET IEFOPZ=(x1,...,xN) MVS command to set/alter configuration
  - Complete replacement of entire configuration, not an append
  - SET IEFOPZ,REFRESH will revert to most recent suffix list
- SMF Type 90 Subtype 38 record contains IEFOPZ configuration
  - Written at IPL and after any SET IEFOPZ updates

Usage & Invocation (cont'd)

- After using the ABO tool to optimize the COBOL libraries, the results should be used to populate the IEFOPZxx configuration
- z/OS will query the IEFOPZ configuration to find out which OLD data sets are to be processed, which members, and which NEW data sets (and for which architecture levels)

Usage & Invocation (cont'd)

- DISPLAY IEFOPZ MVS command to display current configuration
  - Display general status or filtered by one of: DDName, jobname, NEW data set name, OLD data set name, owner
  - Wildcarding * and ? allowed except on STATUS

Ex:

```
D IEFOPZ,STATUS
IEFA160I 16.50.47 IEFOPZ Status
MAXARCH: 11
LNKLST: No   LLA: No
IEFOPZ(s): QV,QW
```

Ex:

```
D IEFOPZ,NEW=JMAN.*
IEFA162I 16.58.54 IEFOPZ New 444
New:        JMAN.NEW1.LOAD
  Old:        JMAN.OLD1.LOAD
    Owner: IBM
New:        JMAN.NEW2.LOAD
  Old:        JMAN.OLD2.LOAD
    Owner: IBM
```

Usage & Invocation (cont'd)

- IEFOPZQ Query Service
  - IEFOPZQ REQUEST=...
    - BY_OLD, BY_NEW, BY_DDJOBNAME, BY_OWNER, STATUSINFO
  - Use IEFOPZAA answer area macro for return info

  Ex: Return NEW for the OLD 'MY.DSN':

```
         IEFOPZQ REQUEST=BY_OLD,                             *
               DSNAME=d,                                     *
               MEMBERS=NO,STATE=ACTIVE,                      *
               ARCH=ar,                                      *
               STATUSINFO=NO,                                *
               ANSAREA=a,ANSLEN=al,                          *
               RETCODE=LRETCODE,RSNCODE=LRSNCODE,            *
               MF=(E,OPTQL)
* Here you would place code to process the return and
* reason codes.
 ...
d        DC    CL44'MY.DSN'
al       DC    A(L'a)
ar       DC    X'FFFF'
DYNAREA  DSECT
a        DS    CL(OPZAAH_LEN+OPZAAOLD_LEN+OPZAANEW_LEN)
LRETCODE DS    F
LRSNCODE DS    F
         IEFOPZQ MF=(L,OPTQL),PLISTVER=MAX
         IEFOPZAA
```

Usage & Invocation (cont'd)

- IEFOPZQ Query Service

Ex: Return matching DDName/Jobname pairs:

```
* Code to set up dd, j, and al and to acquire an answer area
* and place its address into register n
 ...
* Invoke IEFOPZQ
         IEFOPZQ REQUEST=BY_DDJOBNAME,                       *
               DDNAME=dd,JOBNAME=j,                          *
               ANSAREA=(n),ANSLEN=al,                        *
               RETCODE=LRETCODE,RSNCODE=LRSNCODE,            *
               MF=(E,OPTQL)
* Here you would place code to process the return and
* reason codes. If they indicated that not all data was
* returned (reason IEFOPZQRsn_NotAllDataReturned), then
* acquiring a larger answer area, updating the al value and
* retrying IEFOPZQ.
 ...
DYNAREA  DSECT
dd       DS    D
j        DS    D
al       DS    F
LRETCODE DS    F
LRSNCODE DS    F
         IEFOPZQ MF=(L,OPTQL)
```

Usage & Invocation (cont'd)

- LNKLST: If a LNKLST data set is an IEFOPZ-Old, put the IEFOPZ-New data set into the LNKLST just ahead of it
  - LLA will process both IEFOPZ-Old and IEFOPZ-New data sets in the LNKLST
  - Special processing is done if there are multiple IEFOPZ-Old data sets that refer to the same IEFOPZ-New data set
    - Only one IEFOPZ-New ahead of the IEFOPZ-Old data sets
  - On LNKLST COPYFROM, IEFOPZ processing is re-applied to new list
    - Only matters if the configuration changed
- LLA: Outside of the LNKLST, if a library to be managed is an IEFOPZ-Old, also manage the IEFOPZ-New
  - If an update is for an IEFOPZ-Old, also update the IEFOPZ-new
  - Any NEW library that was added to LLA previously remains within LLA management unless explicitly removed

Usage & Invocation (cont'd)

- Batch Allocation: IEFOPZ Processing is done for JOBLIB/STEPLIB DDs

IEFOPZxx:

```
OLDNEW(
  OLD(DSNAME(IEFOPZ.OLDDS))
  New(DSNAME(IEFOPZ.NEWDS) ARCH(10))
  OWNER(IBM)
  ACTIVE
 )
```

JCL:

```
//JOBNAME1 JOB
...
//STEPLIB  DD DSN=IEFOPZ.OLDDS,DISP=(OLD)
//         DD DSN=IEFOPZ.DS2,DISP=(OLD)
```

Output:

```
IEFA170I JOBNAME1 STEP1 STEPLIB - IEFOPZ PROCESSING
CONCATENATED IEFOPZ-NEW IEFOPZ.NEWDS ON VOLUME VOL123
WITH IEFOPZ-OLD IEFOPZ.OLDDS ON VOLUME VOL123
```

Result:

```
//STEPLIB DD DSN=IEFOPZ.NEWDS,DISP=(SHR,KEEP,KEEP)
//        DD DSN=IEFOPZ.OLDDS,DISP=(OLD)
//        DD DSN=IEFOPZ.DS2,DISP=(OLD)
```

Usage & Invocation (cont'd)

- Batch Allocation: IEFOPZ Processing is done for JOBLIB/STEPLIB DDs or for a DDName-Jobname pair match

IEFOPZxx:

```
OLDNEW(
  OLD(DSNAME(IEFOPZ.OLDDS))
  New(DSNAME(IEFOPZ.NEWDS) ARCH(10))
  OWNER(IBM)
  ACTIVE
 )
DDNAME(SYSLMOD) JOBNAME(JOBNAME1)
```

JCL:

```
//JOBNAME1 JOB
...
//SYSLMOD DD DSN=IEFOPZ.OLDDS,DISP=(OLD)
//        DD DSN=IEFOPZ.DS2,DISP=(OLD)
```

Output:

```
IEFA170I JOBNAME1 STEP1 SYSLMOD - IEFOPZ PROCESSING
CONCATENATED IEFOPZ-NEW IEFOPZ.NEWDS ON VOLUME VOL123
WITH IEFOPZ-OLD IEFOPZ.OLDDS ON VOLUME VOL123
```

Result:

```
//SYSLMOD DD DSN=IEFOPZ.NEWDS,DISP=(SHR,KEEP,KEEP)
//        DD DSN=IEFOPZ.OLDDS,DISP=(OLD)
//        DD DSN=IEFOPZ.DS2,DISP=(OLD)
```

Usage & Invocation (cont'd)

- Dynamic Allocation: IEFOPZ Processing is performed when requested via the DALReqIEFOPZ text unit on SVC99
  - Permanently concatenates IEFOPZ-New ahead of IEFOPZ-Old

IEFOPZxx:

```
OLDNEW(OLD(DSNAME(IEFOPZ.OLDDS))
       NEW(DSNAME(IEFOPZ.NEWDS) ARCH(10))
       OWNER(IBM)
       ACTIVE)
```

SVC 99:

```
KEY    #      LEN    PARM
0002   0001   000C   IEFOPZ.OLDDS     DALDSNAM
0078   0000   -      -                DALReqIEFOPZ
```

Usage & Invocation (cont'd)

- Dynamic Allocation: Can return IEFOPZ information
  - Only available on Allocation (S99VRBAL), not Information Retrieval (S99VRBIN)

IEFOPZxx:

```
OLDNEW(OLD(DSNAME(IEFOPZ.OLDDS))
       NEW(DSNAME(IEFOPZ.NEWDS) ARCH(10))
       OWNER(IBM)
       ACTIVE)
```

SVC 99:

```
KEY    #      LEN    PARM
0002   0001   000C   IEFOPZ.OLDDS      DALDSNAM
0056   0001   002C   -                 DALRTDSN
0078   0000   -      -                 DALReqIEFOPZ
007B   0001   1      -                 DALRetInfo
007C   0001   002C   -                 DALRetIEFOPZnewDSN
007D   0001   0006   -                 DALRetIEFOPZnewVol
```

Output:

```
KEY    #      LEN    PARM
0056   0001   000C   IEFOPZ.OLDDS      DALRTDSN
007B   0001   1      '80'X             DALRetInfo
007C   0001   000C   IEFOPZ.NEWDS      DALRetIEFOPZnewDSN
007D   0001   0006   VOL123            DALRetIEFOPZnewVol
```

Interactions & Dependencies

- Software Dependencies
  - None
- Hardware Dependencies
  - None
- Exploiters
  - IBM Automatic Binary Optimizer
  - Customers using infrastructure to facilitate migration from COBOL V4 to COBOL V5

Migration & Coexistence Considerations

- None

Installation

- For z/OS V2R3, no special installation steps are needed
- For z/OS V2R2, APAR OA47689 must be installed
- For exploitation of this function, new IEFOPZxx parmlib member containing IEFOPZ configuration must be defined

Session Summary

- Changes were made in z/OS to provide IEFOPZ infrastructure that the Automatic Binary Optimizer product will exploit to help with COBOL optimization
- IEFOPZ processing via IEFOPZxx parmlib definition and support in Allocation and CSV to build concatenations of “NEW” ahead of “OLD” at appropriate times
  - LLA and LNKLST considerations
  - Batch Allocation via DDName-Jobname pairs
  - Dynamic Allocation via DALReqIEFOPZ
- Avoid JCL updates when updating to new library versions

Appendix

- Publications:
  - z/OS MVS Initialization and Tuning Reference
  - z/OS MVS System Commands
  - z/OS MVS Programming: Assembler Services Reference IAR-XCT
  - z/OS MVS Programming: Authorized Assembler Services Reference ALE-DYN
  - z/OS MVS System Messages, Vol 8 (IEF-IGD)
  - IBM Automatic Binary Optimizer for z/OS 1.1.0