



Assicurazioni Generali HALDB Migration From PDB to HALDB Hints and Tips



The Team

Fiorese.Giampaolo@generali.it

Pettenu.Massimo@generali.it

Piaggi.Fabrizio@generali.it

Polese.Federica@generali.it

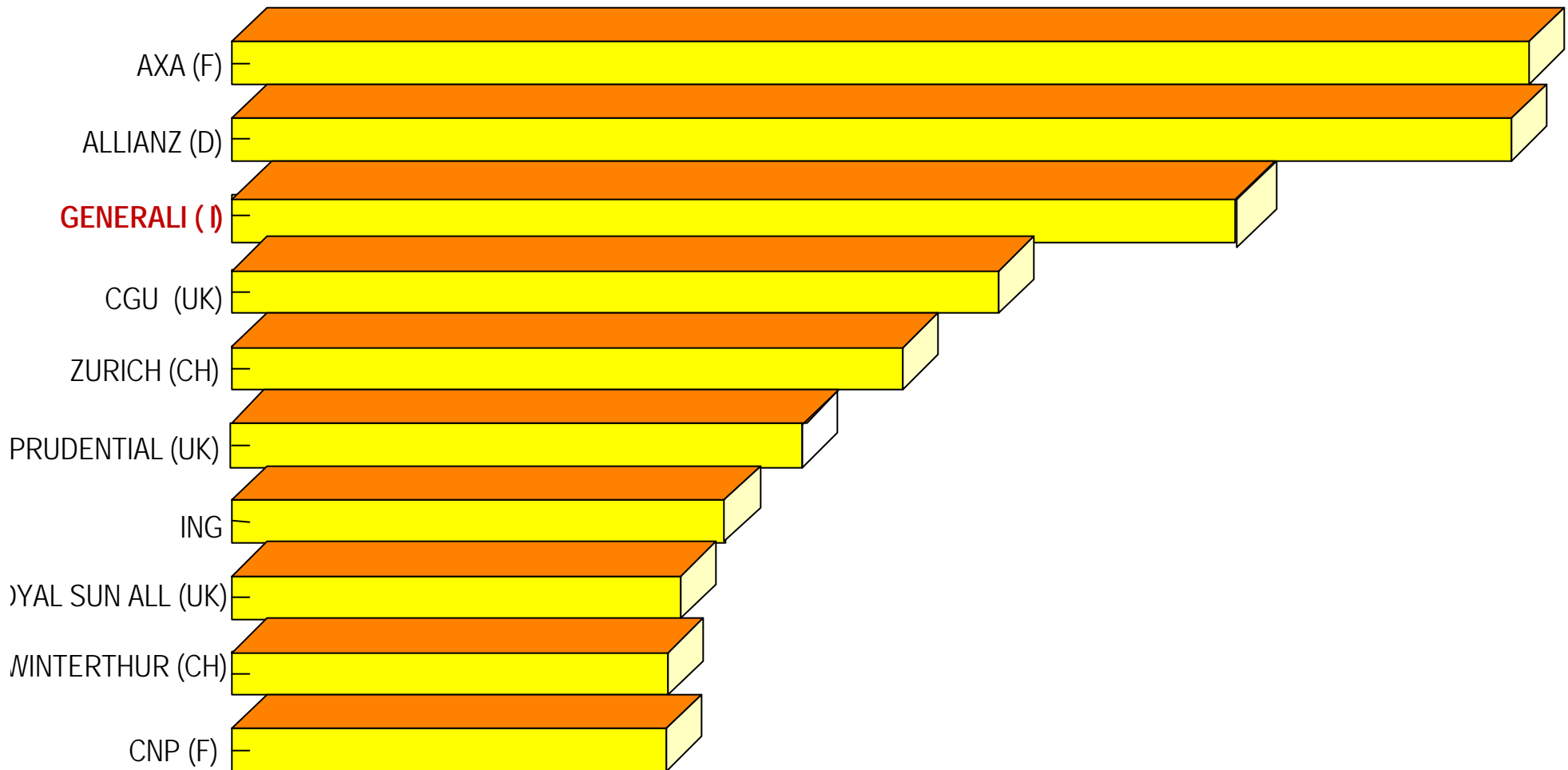
Marina_Canestrelli@it.ibm.com





Mario Rossi/Photo Researchers, Inc.

Main European Insurance Groups



HALDB Migration

From PDB to HALDB Hints & Tips

DATA BASE PARTITIONING

- First implementation 11/1997 on IMS Version 5.1
- Migration from Data Set Group to DB Partitioning
- A two segments structure hit the 4GB on a single SW compressed segment
- >15 Partitioned DBs currently in production on IMS Version 7.1
- Current biggest one > 15GB on 9 partitions
- Quite stable since beginning
- Some problems with an OEM DL1 Utility vendor
 - Stable on 1Q/1998)

GENERALI GROUP INFORMATION SYSTEM



HALDB Migration

From PDB to HALDB Hints & Tips

Why HALDB?

- **Limitations on PDB implementation Version 1**
 - No Primary Index Partitioning
 - No Secondary Index Partitioning (available with PDB Version 2)
 - Low concurrency
 - **NOT AN IBM CODE**
 - No parallel IC/REORG/RECOVER
 - Secondary Indexes still need to be recreated after Reorg
- **Availability & Commands by Partition not by Data Base**
- **New IMS/DB implementation only with HALDB**

March 26, 2001



HALDB Migration

From PDB to HALDB Hints & Tips

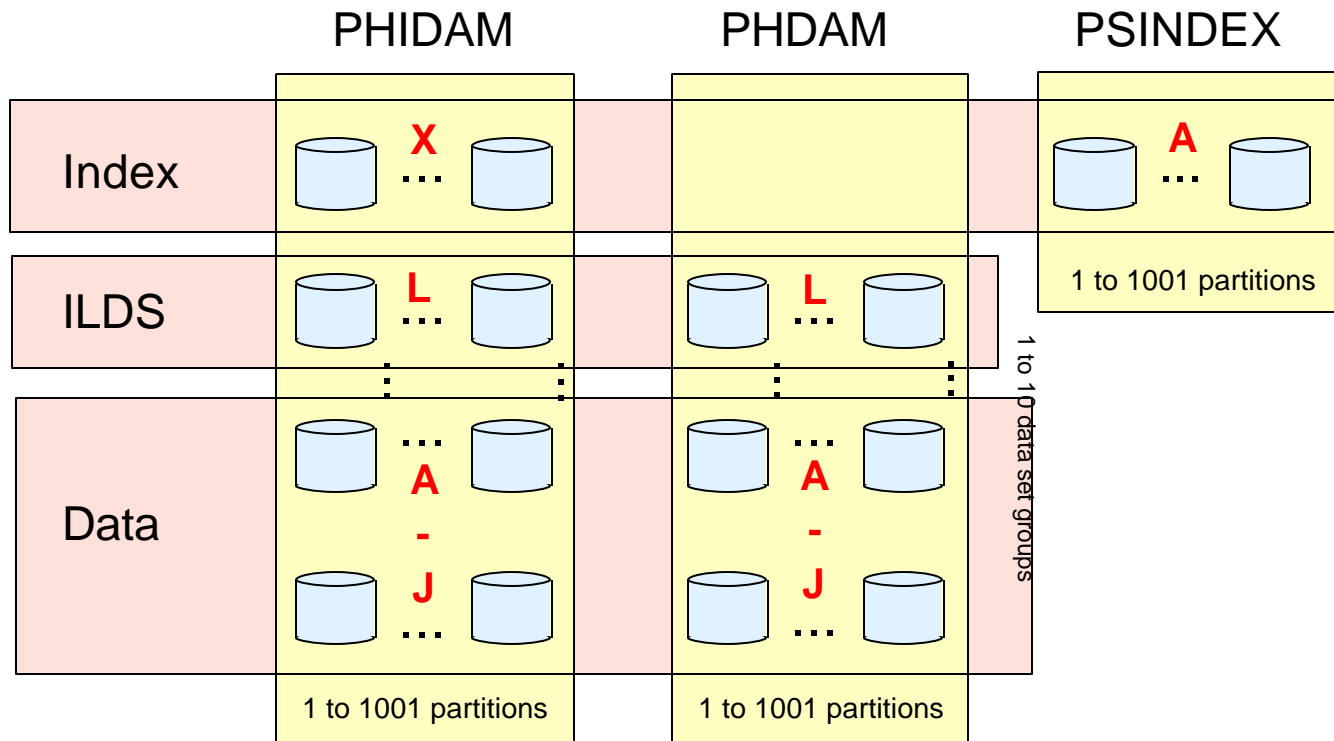
Issues on HALDB

- **MIGRATE vs MIGRATX**
- **Self Healing Pointer process still to be fully understood**
 - How much does it cost?
 - Overhead needs to be measured
- **Backup and Operation processes have to be revised**
 - Secondary Indexes **MUST** be Image Copied again (if no tools used)
 - /DBR and /STA DB commands issued by AOI interface should be revised
- **Education & Training for Application Programmers**
 - They have likely no knowledge on DBRC
 - Usage of Batch Definition will help vs PDU ISPF Interface
- **No OEM vendor support available (planned by 2Q2001)**

GENERALI GROUP INFORMATION SYSTEM



Database Data Sets



The data sets in a partition have generated data set names and DDNAMEs. Letters are used to distinguish them.

X - PHIDAM index

L - ILDS

A through J - Data data sets

A - PSINDEX

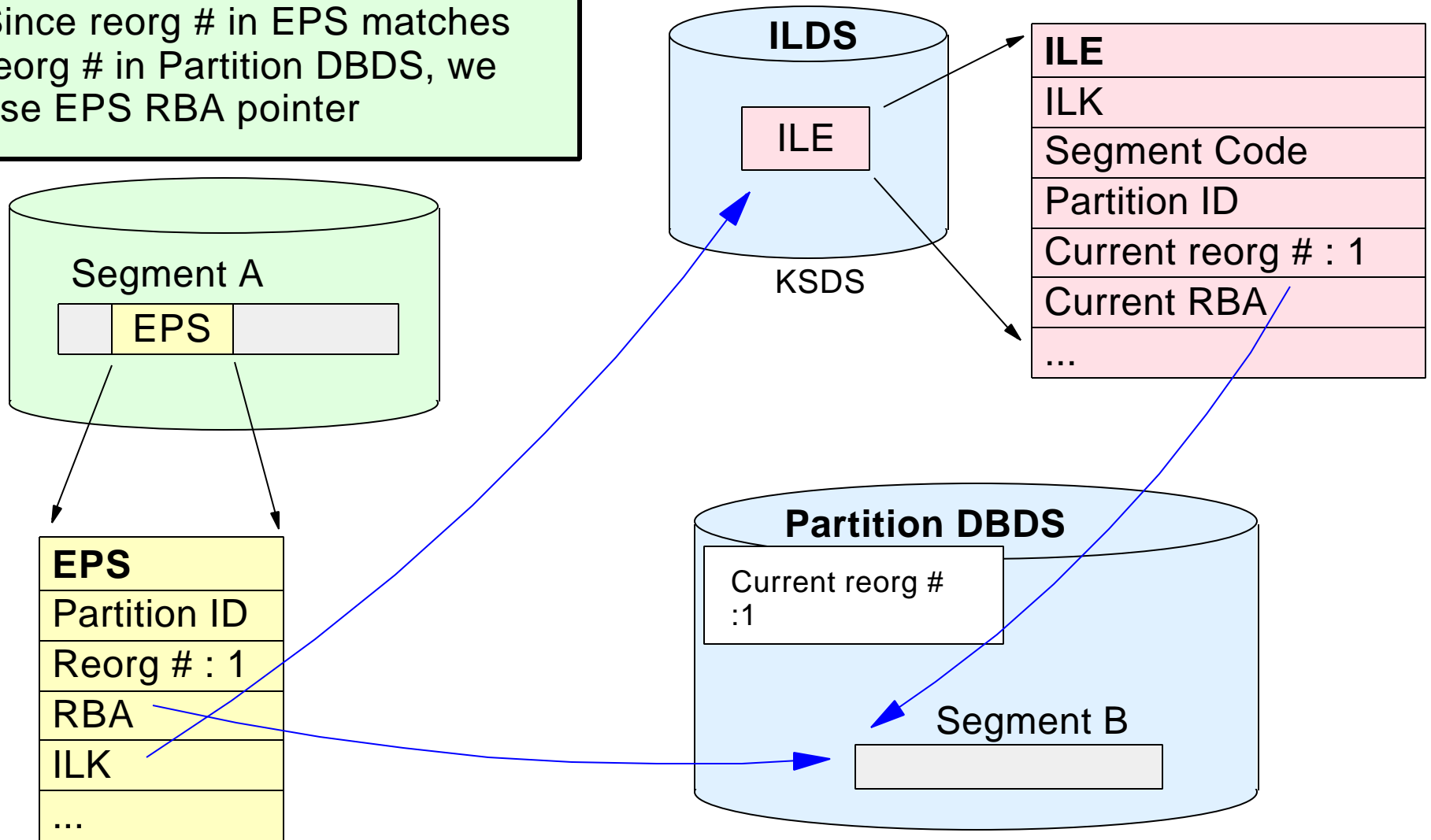
Extended Pointer Set

- **Extended Pointer Set (EPS) is used for logical relationships and secondary indexes**
 - EPS is not updated by reorganizations!
 - EPS contains direct pointer, reorganization number, and ILK
 - If reorg number is current, direct pointer is used
 - If reorg number is not current, ILK is used to find ILE in ILDS
 - ILE contains pointer to segment
 - Direct pointer and reorg number in EPS are updated when ILE is used
- **Self healing pointers!**

Self-Healing Pointers

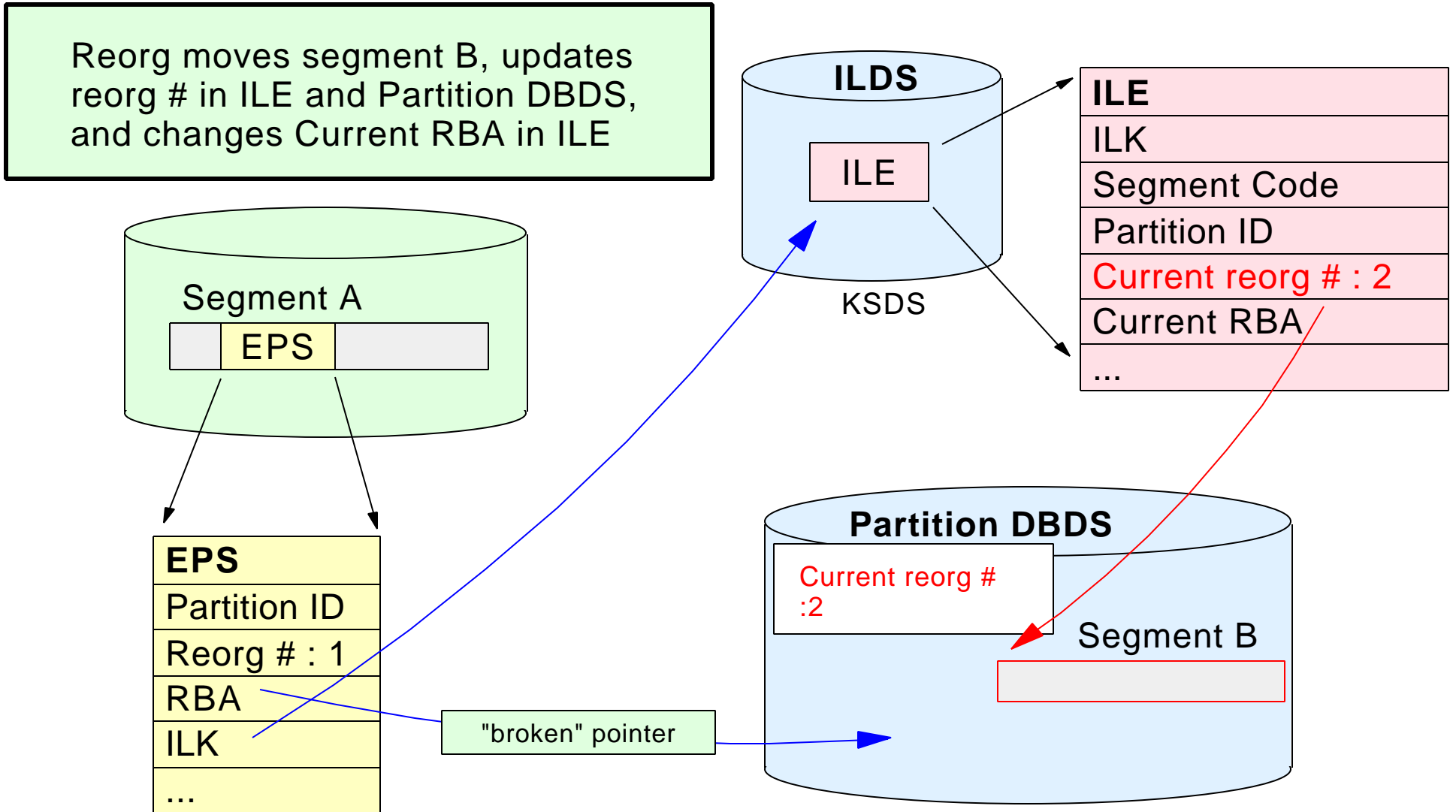
Using an Extended Pointer Set (EPS)

Since reorg # in EPS matches reorg # in Partition DBDS, we use EPS RBA pointer



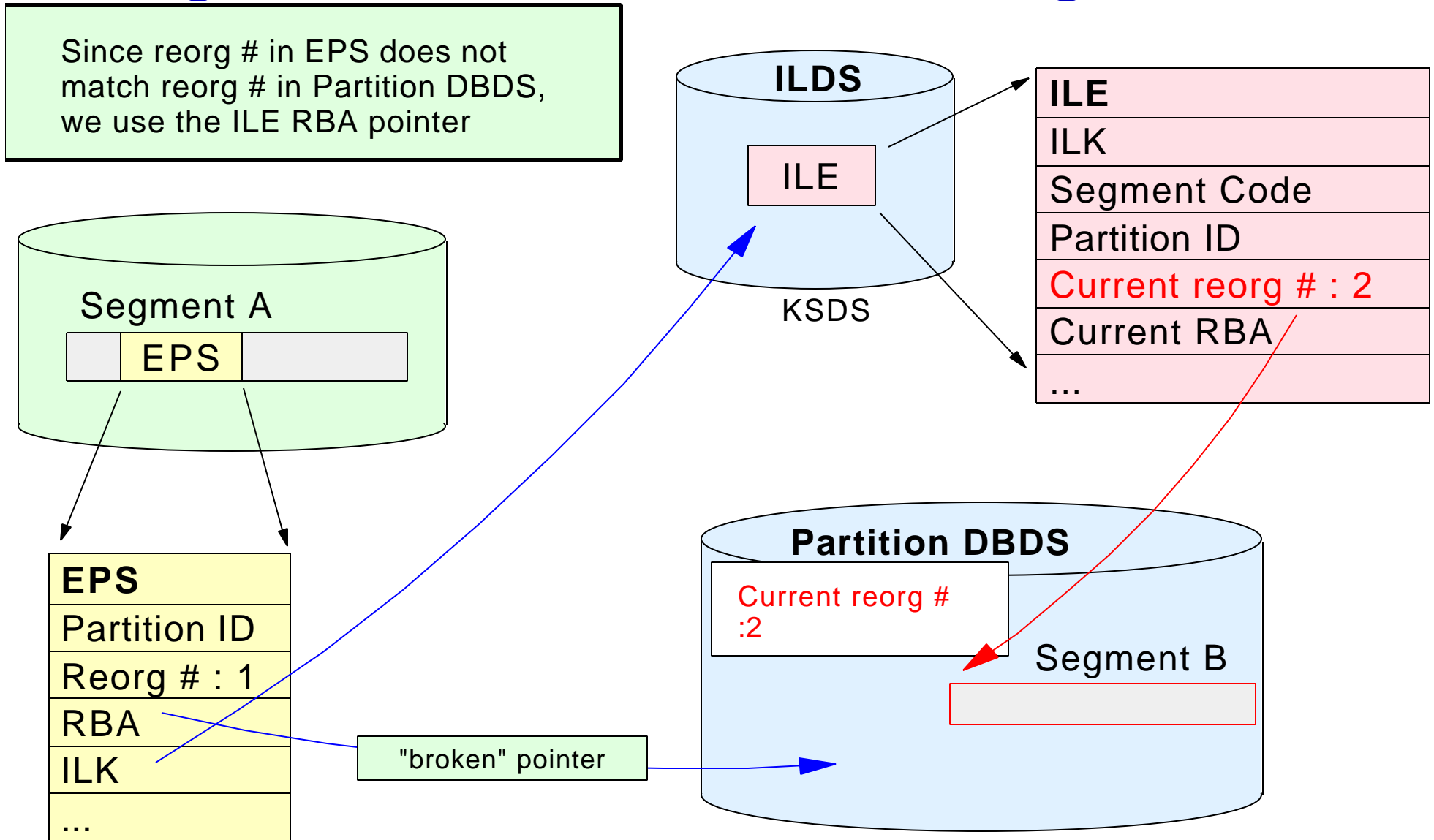
Self-Healing Pointers

After reorganization of Partition



Self-Healing Pointers

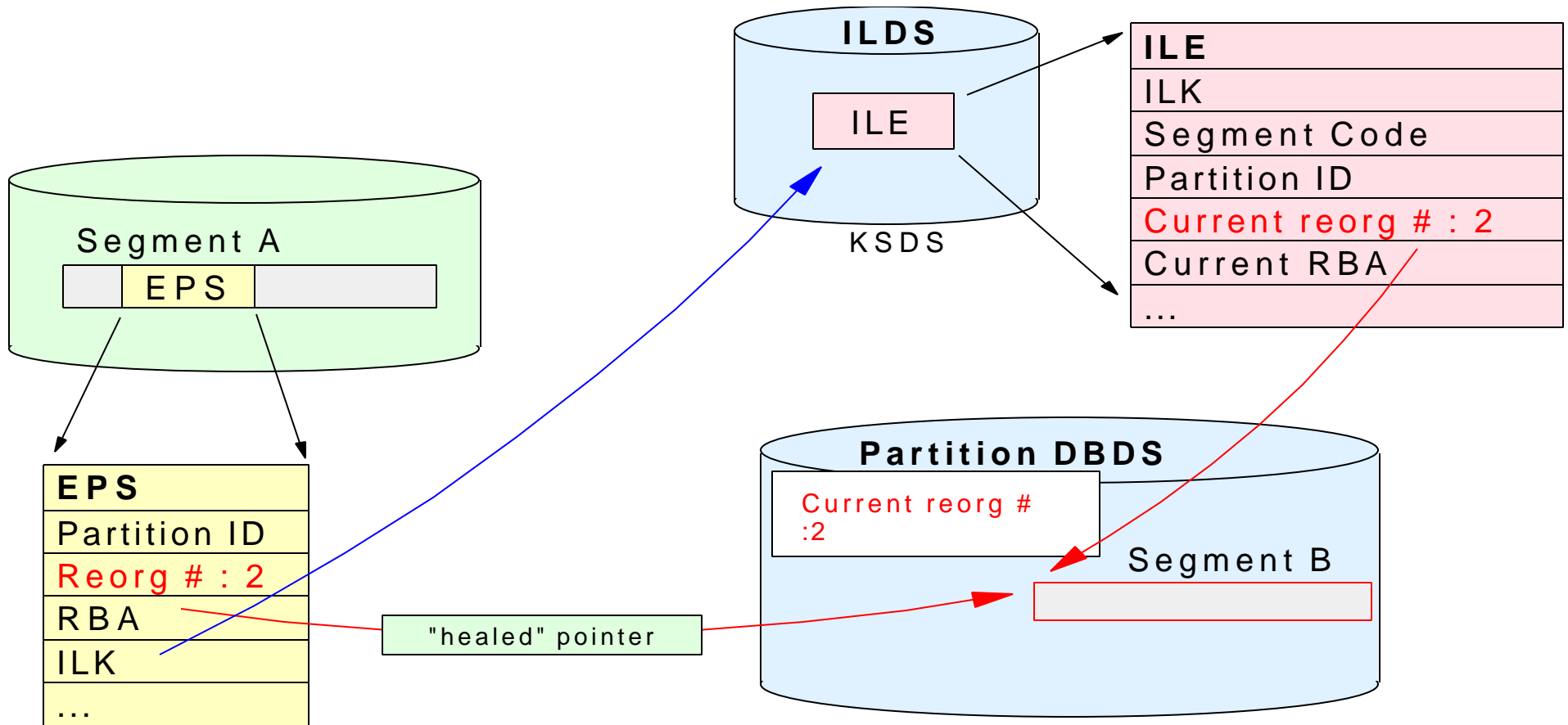
Using the EPS after the reorganization



Self-Healing Pointers

"Healing" the EPS

When we use the ILE, we update the reorg # and RBA pointer in the EPS





HALDB Migration

From PDB to HALDB Hints & Tips

- Migration Aid Utility
 - Cannot be used for Secondary Indexes
 - Three different sysin input
 - Number of Partition wanted
 - Highest Key in Partition
 - Number of Bytes in every Partition
- Logically related DB must all be migrated at the same time
 - HALDB doesn't support logical relationship with FF DB
 - Secondary Indexes must be migrated in parallel



HALDB Migration

From PDB to HALDB Hints & Tips

- Unload with Migrate Option
 - Which one ?

	PDB	FF	PDB+IS	FF+IS	PDB+IS One Step	FF+IS One Step		
MIGRATE	Y	Y	N	Y**	N	N		
MIGRATEX	N	N	N	N	Y*	Y*		
** Slow Performance								
* High Performance								

- **Primary Index is NOT migrated as It will be automatically created by RELOAD**
- **Choose MIGRATE if you don't have Secondary Indexes**
- **Choose MIGRATEX if you have Secondary Indexes**

```

// * IMS 7.1.0
// * UNLOAD DATABASE UTILITY - DFSURGU0
// * UNLOAD DATA PORTION PARTITIONS + SECONDARY INDEX
// * FOR MIGRATION TO HALDB
// *****
// *****
// ULU EXEC PGM=DFSRR00,PARM='ULU,DFSURGU0,ITBTABEL'
// STEPLIB DD DSN=TESTPP.SHRINK7.LOAD,DISP=SHR
// DD DSN=TESTPP.SHRINK7.FDTLIB,DISP=SHR
// DD DSN=TESTIMS.IMS7.RESLIB,DISP=SHR
// DD DSN=TESTIMS.IMS7.DYNALLOC.RESLIB,DISP=SHR
// DFSRESLB DD DSN=TESTIMS.IMS7.RESLIB,DISP=SHR
// IMS DD DSN=TESTIMS.IMS7.DBDLIB,DISP=SHR
// SYSPRINT DD SYSOUT=*
// SYSUDUMP DD SYSOUT=*
// *****
// * DFSURGU1 = DATA PORTION OUTPUT DATA SET
// *****
// DFSURGU1 DD DSN=TESTIMS.IMS710.UNLOAD.ITB.PREHA,DISP=OLD
// *****
// *****
// ITBTABPD DD DSN=TESTIMS.IMS710.ITBTABPD,DISP=SHR ← == Partition 1
// ITBTABP2 DD DSN=TESTIMS.IMS710.ITBTABP2,DISP=SHR ← == Partition 2
// ITBTABIP DD DSN=TESTIMS.IMS710.ITBTABIP,DISP=SHR ← == Primary Index
// ITBTABIC DD DSN=TESTIMS.IMS710.ITBTABIC,DISP=SHR ← == Second. Index
// *****
// * DFSWRK01 DD = SECONDARY INDEX OUTPUT DATA SET
// *****
// DFSWRK01 DD DSN=TESTIMS.IMS710.UNLOAD.ITB.DFSWRK01,
// DISP=(NEW,KEEP),UNIT=3390,VOL=SER=TESTP2,
// SPACE=(CYL,(30,30))
// *****
// * DFSSRT01 DD = SORT STATEMENT FOR SECONDARY INDEX
// *****
// DFSSRT01 DD DSN=TESTIMS.IMS710.UNLOAD.ITB.DFSSRT01,
// DISP=(NEW,KEEP),UNIT=3390,VOL=SER=TESTP2,
// SPACE=(TRK,(10,10))
// *
// DFSVSAMP DD DISP=SHR,DSN=TESTIMS.IMS7.INSTLIB(DFSVSAM2)
// SYSIN DD *
MIGRATX=YES
// *

```


HALDB Migration

From PDB to HALDB Hints & Tips

- **MIGRATEX support has been delivered through maintenance by **APAR PQ37015 PTF UQ47416** after GA**
- **No documentation available in regular manual**
- **Up to now documentation is available only in PTF cover Letter**

```

//TVIT91RE JOB ,CLASS=R,MSGCLASS=Z,REGION=6M,NOTIFY=&SYSUID
//*****
//* IMS 7.1.0
//* RELOAD DATABASE UTILITY - DFSURGL0 - ITBTABEL + SECONDARY INDEXES
//* TO BE USED FOR RELOAD HALDB AHTER MIGRATION
//*****
//*
//*****
//* RELOAD DATA PORTION *
//*****
//*
//* ULU,DFSURGL0,.....(DBDNAME DB TO BE LOADED)
//*****
//ULU EXEC PGM=DFSRR00,PARM='ULU,DFSURGL0,ITBTABEL'
//STEPLIB DD DSN=TESTIMS.IMS7.RESLIB,DISP=SHR
// DD DSN=TESTPP.SHRINK7.LOAD,DISP=SHR
// DD DSN=TESTPP.SHRINK7.FDTLIB,DISP=SHR
//DFSRESLB DD DSN=TESTIMS.IMS7.RESLIB,DISP=SHR
//IMS DD DSN=TESTIMS.IMS7.DBDLIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//*****
//* DFSUINPT = INPUT DATASET CREATED BY HD REORG. UNLOAD UTIL.
//*****
//DFSUINPT DD DSN=TESTIMS.IMS710.UNLOAD.ITB.PREHA,DISP=SHR
//*****
//* DFSURWF1 = WORK DATASET PER PREFIX RES. - DUMMY IF PHDAM OR PHIDAM
//*****
//DFSURWF1 DD DUMMY
//*****
//* PARTITIONS DATA SETS ARE DYNAMICALLY ALLOCATED VIA DBRC FOR PHIDAM
//*****
//*
//*****
//* DFSURCDS = OUTPUT DATASET GENERATED BY PREREORG UTILITY (DFSURPR0)
//* DUMMY IF NO LOGICAL RELATIONSHIP
//*****
//DFSURCDS DD DUMMY
//*****
//DFSVSAMP DD DSN=TESTIMS.IMS7.INSTLIB(DFSVSAM2),DISP=SHR
//*

```

```

//*****
//* SECONDARY INDEXES RELOAD *
//*****
//*
//SORTX EXEC PGM=SORT
//SORTIN DD DSN=TESTIMS.IMS710.UNLOAD.ITB.DFSWRK01,DISP=(OLD,PASS)
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(50,10),,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(50,10),,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(50,10),,CONTIG)
//SYSOUT DD SYSOUT=*
//SYSIN DD DSN=TESTIMS.IMS710.UNLOAD.ITB.DFSSRT01,DISP=OLD
//SORTOUT DD DSN=TESTIMS.IMS710.UNLOAD.ITB.SORTOUT,DISP=(,PASS),
// UNIT=3390,VOL=SER=IMSPR3,SPACE=(CYL,(40,30))
//*
//ULUX EXEC PGM=DFSRR00,PARM='ULU,DFSURGL0,ITBTABIC'
//STEPLIB DD DSN=TESTIMS.IMS7.RESLIB,DISP=SHR
// DD DSN=TESTIMS.IMS7.DYNALLOC.RESLIB,DISP=SHR
// DD DSN=TESTPP.SHRINK7.LOAD,DISP=SHR
// DD DSN=TESTPP.SHRINK7.FDTLIB,DISP=SHR
//DFSRESLB DD DSN=TESTIMS.IMS7.RESLIB,DISP=SHR
//IMS DD DSN=TESTIMS.IMS7.DBDLIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//*****
//* DFSUINPT = INPUT DATASET CREATED BY SORT IN PREVIOUS STEP
//*****
//DFSUINPT DD DSN=TESTIMS.IMS710.UNLOAD.ITB.SORTOUT,DISP=OLD
//*****
//* DFSURWF1 = WORK DATASET FOR PREFIX RES. - DUMMY FOR PHDAM OR PHIDAM
//*****
//DFSURWF1 DD DUMMY
//*****
//* PARTITIONS DATA SETS ARE DYNAMICALLY ALLOCATED VIA DBRC FOR PHIDAM
//*****
//*
//*****
//* DFSURCDS = OUTPUT DATASET GENERATED BY PREREORG UTILITY (DFSURPR0)
//* DUMMY IF NO LOGICAL RELATIONSHIP
//*****
//DFSURCDS DD DUMMY
//*
//DFSVSAMP DD DSN=TESTIMS.IMS7.INSTLIB(DFSVSAM2),DISP=SHR
//*
//*

```



HALDB Migration

From PDB to HALDB Hints & Tips

- Save somehow your Recon Information in case of Fallback if you are going to have the same DB NAME
 - Requirement under evaluation by SVL
- Evaluate have the old DB defined as Logical DB if you choose to have a different HALDB dbname
- Delete Recon Information
- Delete Primary Index
 - From Sysgen, from DBDLIB
- Remove DFSMDA member from dynalloc library
- Modify DBD to be an HALDB DBD
 - Remove LCHILD Primary Index Statement
 - Add RKSIZE parameter in the LCHILD Secondary Index DBDs
 - RKSIZE is the KEY length of the Target DB Segment
- Modify any program written to process a Secondary Index as a standalone DB
 - PSINDEX has an 8 Bytes prefix instead of 4
 - /SX 4 Bytes
 - ILK 8 Bytes (RBA+PID+RN)

DB U7BTABEL before migration to HALDB

HIDAM

```

DBD      NAME=U7BTABEL,ACCESS=(HIDAM,VSAM)
DD1      PART DD1=U7BTABEL,HIGHKEY='12345.....'
DD2      PART DD2=U7BTABE2
          SEGMENT,NAME=SEGTABEL,PTR=NT,
          BYTES=(4070,26)
FIELD NAME=(KEYTABEL,SEQ,U),BYTES=24,START=3
LCHILD NAME=(SEGMKEYT,U7BTABIP),PTR=INDX
LCHILD NAME=(SEGISCON,U7BTABIC),PTR=INDX
XDFLD NAME=CONFANA,SEGMENT=SEGCONFU,
SRCH=KEYCONFU,SUBSEQ=/SXCONT
SEGM NAME=SEGCONFU,
PARENT=((SEGTABEL,SNGL)),PTR=T,
BYTES=25
FIELD BYTES=25,START=1,NAME=(KEYCONFU,SEQ,U)
FIELD NAME=/SXCONT
DBDGEN
FINISH
END
    
```

PRIMARY INDEX

```

DBD      NAME=U7BTABIP,ACCESS=(INDEX,VSAM)
DATASET DD1=U7BTABIP,DEVICE=3380
SEGM NAME=SEGMKEYT,BYTES=24
FIELD NAME=(CHIAVETB,SEQ,U),BYTES=24,START=
LCHILD NAME=(SEGTABEL,U7BTABEL),
INDEX=KEYTABEL
DBDGEN
FINISH
END
    
```

SECONDARY INDEX

```

DBD      NAME=U7BTABIC,ACCESS=INDEX
DATASET DD1=U7BTABIC,DEVICE=3380
SEGM NAME=SEGISCON,BYTES=29
FIELD NAME=(KKKCON,SEQ,U),BYTES=29,START=1
FIELD NAME=CONISKEY,BYTES=25,START=1
LCHILD NAME=(SEGTABEL,U7BTABEL),
INDEX=CONFANA
DBDGEN
FINISH
END
    
```

GENERALI GROUP INFORMATION SYSTEM



PDB after migration to HALDB

PHIDAM

```
DBD NAME=U7BTABEL,ACCESS=(PHIDAM,VSAM)
HALDB DATASET DD1=U7BTABEL,DEVICE=3380
SEGM NAME=SEGTABEL,PTR=NT, X
    BYTES=(4070,26)
FIELD NAME=(KEYTABEL,SEQ,U),BYTES=24,START=3
LCHILD NAME=(SEGISCON,U7BTABIC),PTR=INDX
XDFLD NAME=CONFANA,SEGMENT=SEGCONFU, X
    SRCH=KEYCONFU,SUBSEQ=/SXCONT
SEGM NAME=SEGCONFU,PARENT=((SEGTABEL,SNGL)),PTR=T, X
    BYTES=25
FIELD BYTES=25,START=1,NAME=(KEYCONFU,SEQ,U)
FIELD NAME=/SXCONT
DBDGEN
FINISH
END
```

PSINDEX

```
DBD NAME=U7BTABIC,ACCESS=(PSINDEX,VSAM)
SEGM NAME=SEGISCON,BYTES=33
FIELD NAME=(KKKCON,SEQ,U),BYTES=33,START=1
FIELD NAME=CONISKEY,BYTES=25,START=1
LCHILD NAME=(SEGTABEL,U7BTABEL), X
INDEX=CONFANA,RKSIZE=24
DBDGEN
FINISH
END
```



HALDB Migration

From PDB to HALDB Hints & Tips



- **Execute DBDGEN**

HALDB Batch Initialization

HALDB Batch init support has been delivered through maintenance after GA

- **Can be used instead of ISPF PDU application**
 - **Easier approach for application programmers as it requires less knowledge on DBRC**
 - **Must be run as a standard batch job**
- **No modification to HALDB are allowed via Batch PDU**
- **No HALDB entries can be deleted via Batch PDU**
- **PDU must be used for such changes**
- **INIT HALDB DFSUNPT0 or DFSURPR0**

HALDB Migration

From PDB to HALDB Hints & Tips



HALDB Batch Initialization

```
//PDU          EXEC PGM=DSPURX00
//STEPLIB      DD DSN=IMS.RESLIB,DISP=SHR
//            DD DSN=IMS.DYNALLOC.RESLIB,DISP=SHR
//IMS         DD DSN=IMS.DBDLIB,DISP=SHR
//SYSPRINT    DD SYSOUT=*
//SYSIN       DD *
              INIT.DB DBD(HALDBXX) RECOVABL SHARELVL(2) -
              TYPHALDB
              INIT.PART DBD(HALDBXX) PART(HALDB1) -
              KEYSTRNG(X'C7E9E9E9E9E9E9E9E9E940') -
              DSNPREFIX(IMS.HALDBXX) GENMAX(2)
              INIT.PART DBD(HALDBXX) PART(HALDB2) -
              KEYSTRNG(X'E97E9E9E9E9E9E9E9E9E940') -
              DSNPREFIX(IMS.HALDBXX) GENMAX(2)
```


HALDB Migration

From PDB to HALDB Hints & Tips

- **BATCH HALDB INIT support has been delivered through maintenance by **APAR PQ35893 - PTF UQ49705** after GA**
- **No documentation available in regular manual**
- **Up to now documentation is available only in PTF cover Letter**



HALDB Migration

From PDB to HALDB Hints & Tips

- Reload can be a single multiple process
 - Related to migration unload process
- Which situation after First Reload?
 - IC Needed Flag set at DBDS Level
 - Partition must be ImageCopied
 - In case of Logical Relationship ALL EPS have a Broken Pointer
 - ILDS must be accessed
- Consider running a Dummy Process (BMP or Batch DL1) before resuming normal Online and Batch processes to self-heal all the broken pointers
- Same could apply after normal Reorg



HALDB Migration From PDB to HALDB Hints & Tips

	NON HALDB	HALDB			
	PROCOPT=G & PROCSEQ	First Run after migration PROCOPT=G & PROCSEQ	First Run with PROCOPT=A PROCSEQ	Second run with PROCOPT=A & PROCSEQ	Second run with PROCOPT=G & PROCSEQ
Elapsed	4,1	8,4	12,8	6,5	6,5
CPU	0,9	1,8	2	1,3	1,3
CPU Increment	---	100%	122%	44%	44%

BMP to read about 320.000 Segments through Secondary Index

Legenda	
CPU seconds	
ELAPSED minutes	

HALDB Migration

From PDB to HALDB Hints & Tips

- Reload can defer ILDS creation if NOILDS options used (ILDSMULTI is the default)
 - ILDS left with Recovery Needed Flag ON
- ILDS must be rebuilt immediately after by DFSPREC0
- DB must be imagecopied after reload
- Deferred ILDS creation in the RELOAD process has been delivered through maintenance by **APAR PQ36991 - PTF49987** beginning of February 2001
- **No documentation available in regular manual**
- **Up to now documentation is available only in I cover Letter**

GENERALI GROUP INFORMATION SYSTEM



HALDB Migration

From PDB to HALDB Hints & Tips



RELOAD HALDB AFTER MIGRATE

Reload with NOILDS		Reload with ILDSMULTI		NOILDS vs ILDSMULTI		NOILDS + ILDS REBUILD vs ILDSMULTI	
CPU	Elapsed	CPU	Elapsed	CPU	Elapsed	CPU	Elapsed
1,3	12,4	1,9	14,3	-46,2%	-15,3%	20,8%	5,7%

Legenda	
CPU	minutes
Elapsed	minutes

	RECOVERY ILDS	
ILDS PART 1 Rebuild	0,3	3,6
ILDS PART 2 Rebuild	0,2	2,5
ILDS PART 3 Rebuild	0,2	2,5
ILDS PART 4 Rebuild	0,2	1,7
ILDS PART 5 Rebuild	0,2	3,5

DB Info	
DBDNAME	U7BTABEL
Number of partitions	5
Total Number of Segments	1.614.631
Organization Type	PHIDAM
Part 1 segment #	273.752
Part 2 segment #	380.950
Part 3 segment #	338.284
Part 4 segment #	277.830
Part 5 segment #	343.815



HALDB Migration

From PDB to HALDB Hints & Tips

» **CONCLUSION**

- **Migration Process from FF to HALDB is safe and easy**
 - Thoroughly test the **FALLBACK** Scenario before Production
- Carefully plan the number of partitions as it has impacts on normal operations and backup strategies
- Several options available for Migration and Reload
 - Migrate, Migratex, NOILDS, ILDSMULTI
- Evaluate running a dummy batch job after Reload/Reorg with UPDATE procopt
- HALDB Batch Performance are a little worse than traditional Full Function but
 - New HALDB Features **outperform** and **highly compensate** some batch performance drawbacks

HALDB Migration

From PDB to HALDB Hints & Tips



Bibliografy

www.redbooks.ibm.com

IMS Version 7 High Availability

Large Database Guide

SG24-5751-00

IMS Version 7 Release Guide

SG24-6753-00

www.software.ibm.com/data/ims



Assicurazioni Generali

Thanks to everybody



The Team

Fiorese.Giampaolo@generali.it

Pettenu.Massimo@generali.it

Polese.Federica@generali.it

Piaggi.Fabrizio@generali.it

Marina_Canestrelli@it.ibm.com