Autonomics for IMS

with the IBM Management Console for IMS and DB2 for z/OS

Janet LeBlanc
IMS Tools Product Manager and Development Strategist

March 16, 2015

Session A01
IMS Technical Symposium 2015

IBM

Autonomics

Autonomic computing refers to the self-managing characteristics of distributed computing resources, adapting to unpredictable changes while hiding intrinsic complexity to operators and users. Started by IBM in 2001, this initiative ultimately aims to develop computer systems capable of self-management, to overcome the rapidly growing complexity of computing systems management, and to reduce the barrier that complexity poses to further growth.
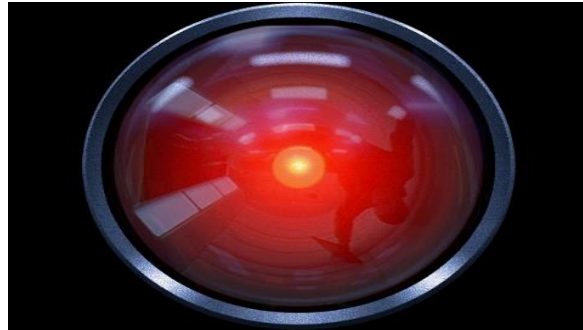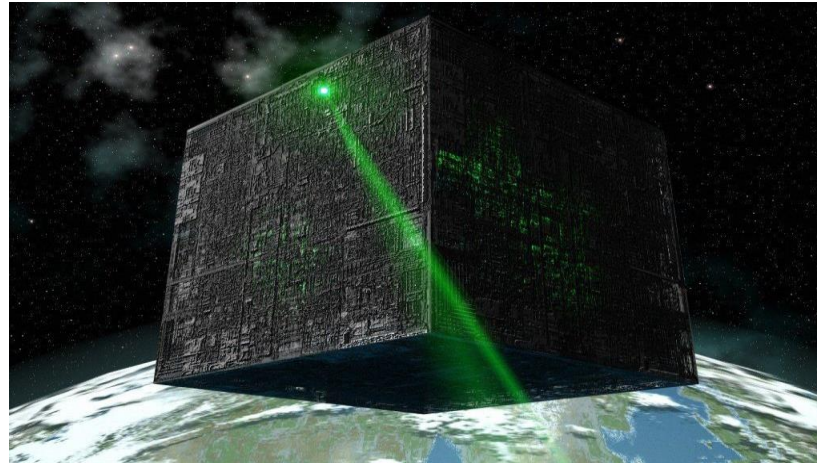

Source: Wikipedia, Oct 2014, http://en.wikipedia.org/wiki/Autonomic_computing

The Inevitability of Autonomics

Forecasts suggest that the number of computing devices in use will grow at 38% per year [1] and the average complexity of each device is increasing. [1] Currently, this volume and complexity is managed by highly skilled humans; but the demand for skilled IT personnel is already outstripping supply, with labour costs exceeding equipment costs by a ratio of up to 18:1… [2]

Source: Wikipedia, Oct 2014, http://en.wikipedia.org/wiki/Autonomic_computing
[1] Horn. "Autonomic Computing:IBM's Perspective on the State of Information Technology"
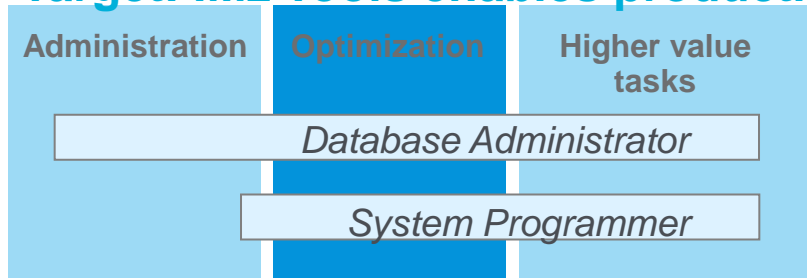[2] **Jump up^** 'Trends in technology', survey, Berkeley University of California, USA, March 2002

The Value of Autonomics

Today

Administration	Optimization	Higher value tasks

Database Administrator

System Programmer
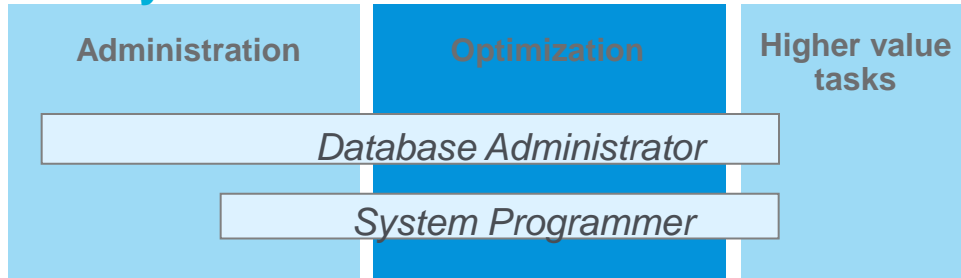
Target: IMz Tools enables productivity

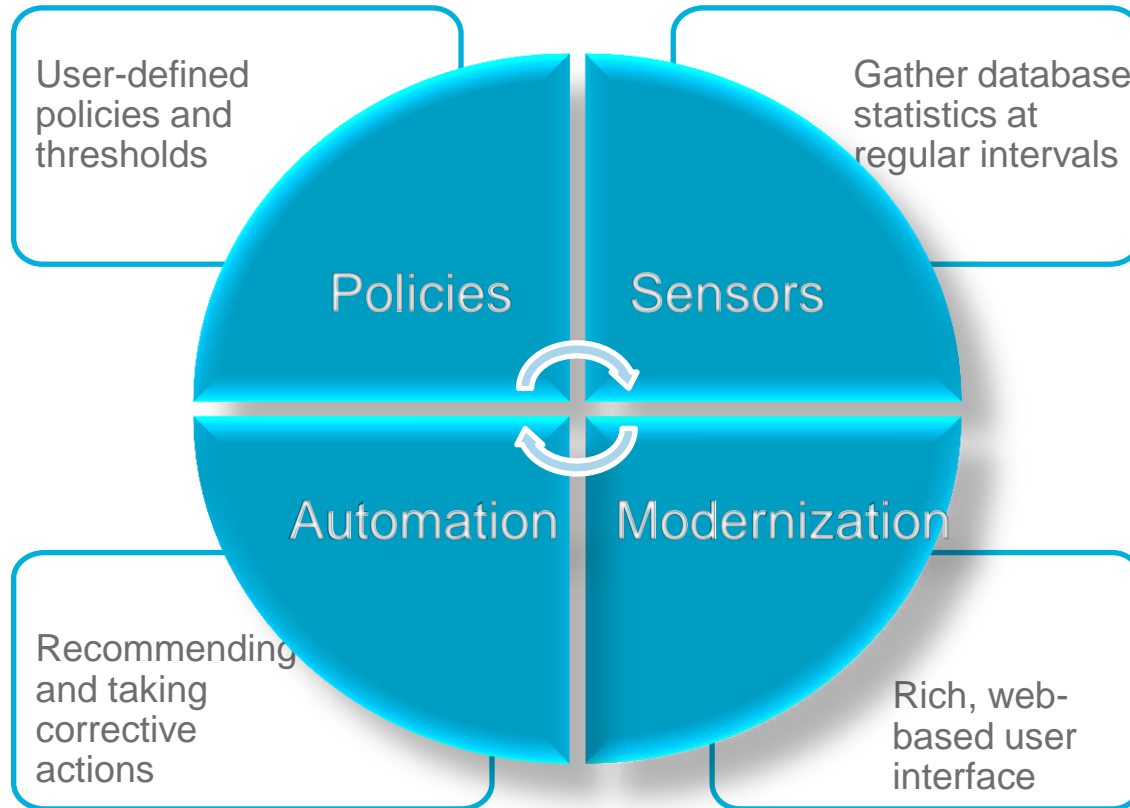| Administration	Optimization	Higher value tasks

Database Administrator

System Programmer

…the essence of autonomic computing is system self-management, delivering better system behavior and **freeing administrators from low-level task management.**

Source: Wikipedia, Oct 2014, http://en.wikipedia.org/wiki/Autonomic_computing

Our Vision – Putting information to work for you

User-defined policies and thresholds

Gather database statistics at regular intervals

Policies

Sensors

Automation

Modernization

Recommending and taking corrective actions

Rich, web-based user interface

IMS Database Solution Pack

- ❖ Autonomics
- ❖ IBM Management Console

- ❖ IMS Online Reorg Facility
- ❖ DB Reorg Expert
  - ❖ Unload
  - ❖ Load
  - ❖ Prefix Resolution / Update
  - ❖ Index Builder
- ❖ HP Image Copy
- ❖ HP Pointer Checker
  - ▪ IMS DB Repair Facility
- ❖ IMS Library Integrity Utilities
- ❖ IMS HALDB Toolkit

IMS Fast Path Solution Pack

- ❖ Autonomics
- ❖ IBM Management Console

- ❖ HP FP Utilities
  - ▪ FP Advanced Utilities
  - ▪ FP Online Utilities
- ❖ IMS DB Repair Facility
- ❖ IMS HP Image Copy
- ❖ IMS Library Integrity Utilities

Sensors: Collecting the Basic Information You Need
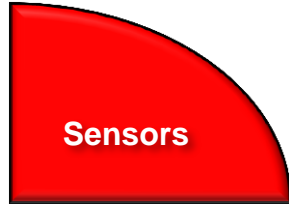
Sensors

- Statistical point-in-time sensor data on your FF/FP Databases
  - Stored in IMS Tools Knowledge Base repository
  - Historically maintained per user specifications
  - Over 60 separate data elements related to space usage, optimization, and fragmentation
    - data set extents, DASD volume usage, data set free space, roots distribution, RAP usage, CI/CA splits, and IMS free space, etc

- Two methods of collection:
  - Standalone database Sensor utilities for full-function and Fast Path databases
  - Integrated with existing IMS Tools utilities

- Integrated Tools support
  - HP Image Copy, HP Pointer Checker, DB Reorg Expert
  - FPA: Reload, Change, Reorg, Analyze
  - FPO: Online PC, Online Reorg

Policies: Using Sensor Data to Make Decisions

Policies
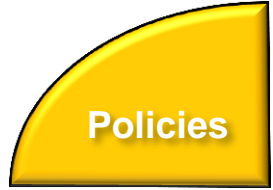
- Policy definitions are used to evaluate specific database states
  - Threshold values are compared against sensor data for a given database or group of databases
  - When thresholds are met or exceeded, exceptions occur

- Works "out of the box"
  - Ships with predefined policies and threshold values
  - Full ISPF interface provided for policy management

- Customizable to fit your shop
  - You can define your own sets of threshold values
  - Customize the messages sent when exceptions do occur
  - Specify who receives which messages and how
    - WTO, e-mail, or text

Automation: Delivering on our Vision

Automation

- IBM Tools Autonomics Director 1.3 (Passive)
  - Automates collection and analysis of Sensor Data
  - Recommends when databases should be reorganized
    - With email or text notifications
  - Provides a scheduling feature that allows you to control how frequently sensor data is collected and how frequently policies are evaluated
  - Flexible scheduling around pre-defined PEAK times

- IBM Tools Autonomic Director 1.4 (Active)
  - Actively initiate recommended actions on user-defined database groups
    - Discovery feature for identifying related database groups
    - Ability to manage and coordinate reorganization of multiple IMS database groups as if reorganizing a single database
    - Ability to manage FP Online Area Extend
    - Flexible scheduling only in pre-defined Maintenance windows

Updates for DB Solution Pack

Automated DB Management Solution by Autonomics Director (AD)
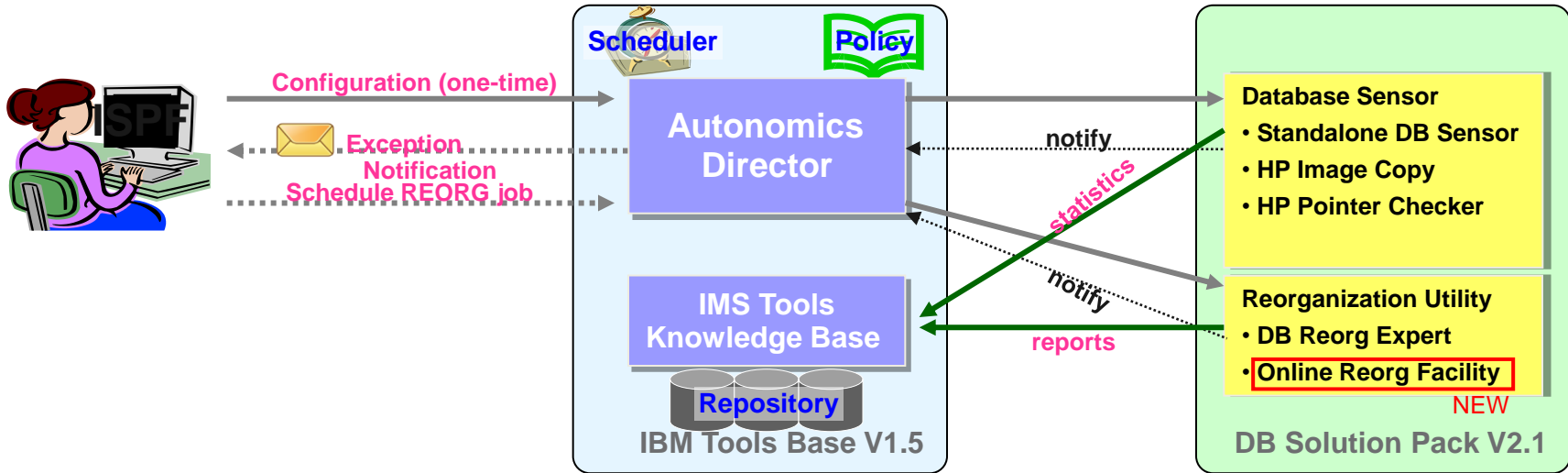
- ✓ Autonomics Director schedules Database Sensor jobs
    - ✓ DB Sensor collects database statistics and stores them in the repository
- ✓ Autonomics Director evaluates the statistics by predefined policies
- ✓ (Passive only) Autonomics Director notifies user if reorganization is needed
- ✓ Autonomics Director schedules Reorganization Utility jobs
    - ✓ Reorganization Utility reorganizes databases and stores reports in the repository
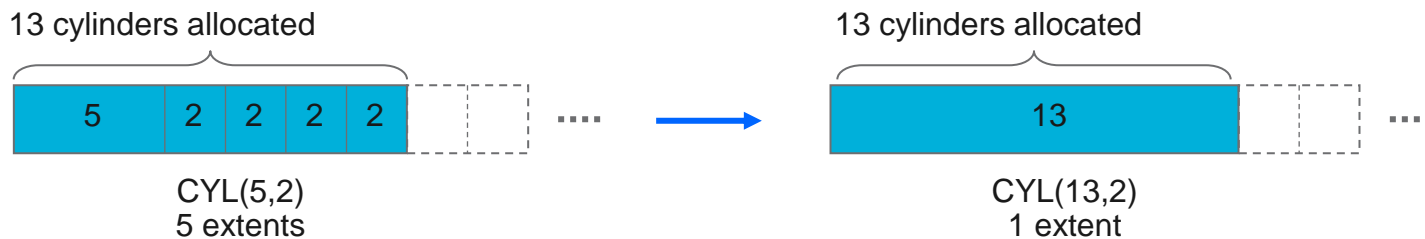
IMS Online Reorganization Facility

✓ Reorganizes an IMS database in a single job step
  ➢ Allocates the required datasets dynamically and invokes the reorganization utilities automatically Unload, Reload, Image Copy, etc.
  ➢ Eliminates the need for manual intervention before, during and after a reorganization

✓ Reduces database downtime significantly
  ➢ Allows a database to be updated while the reorganization process is occurring
  ➢ A short downtime is necessary to alter the database data sets from the shadow names to the original names (Takeover)

✓ Supporting Database Types
  ➢ HDAM, HIDAM, HISAM, SHISAM, Secondary Index, PHDAM, PHIDAM, PSINDEX

✓ Prerequisites
  ➢ IBM Tools Base
    ➢ Tools Online System Interface (TOSI)
    ➢ Generic Exits
    ➢ (IMS Tools Knowledge Base (ITKB) and Autonomics Director are optional)
  ➢ Databases must be registered to DBRC

Increasing the primary allocation size of database data set

✓ SHADOW keyword is enhanced to support a new value
- ➢ Valid values for SHADOW keywords
  - SHADOW(A): Allocate
  - SHADOW(E): Exist
  - SHADOW(R): Allocate and Resize **NEW**
- ➢ Shadow datasets are deleted and defined using the attributes of the original data sets.
- ➢ The allocated size of the original data sets is used for the primary allocation size.

13 cylinders allocated

| 5 | 2 | 2 | 2 | 2 |

CYL(5,2)
5 extents

→

13 cylinders allocated

| 13 |

CYL(13,2)
1 extent

- ➢ Useful when you want to have all the database records stored in the primary extent of the database data sets after the reorganization.
- ➢ Attention: When the original data sets are SMS-managed multi-volume data sets with the guaranteed space attribute, the use of SHADOW(R) can cause the shadow data sets to be larger than necessary because SMS preallocates primary space on all volumes.

Updates for FP Solution Pack

FP Solution Pack utility functions

DEDB Maintenance Task	Solution Function	Fast Path Online (FPO) utility	Online Space Management (OSM) utility	Fast Path Advanced (FPA) utility command	Fast Path Basic (FPB) utility
Managing space	Unload	Online Data Extract (ODE)		UNLOAD	DEDB Unload
	Reload			RELOAD	DEDB Reload
	Shadow area reorganization	* This function is provided by IMS DEDB Alter	Combines OER, OAE, and OPC functions, and supports **Active Autonomics**	CHANGE	
	In-place area reorganization	Online Expert Reorg (OER)		REORG	
	IOVF/SDEP expansion	Online Area Extender (OAE)		RELOAD or CHANGE	
Analyzing space utilization	Integrity verification and space analysis	Online Pointer Checker (OPC)		ANALYZE	DEDB Pointer Checker
	Print DMAC	Online DMAC Print (ODM)		DMACPRT	
Tuning space & randomizer	Area tuning aid	* OPC can produce input for FPA TUNE		TUNE	DEDB Tuning Aid
Extracting segment data	Extract specific segments	Online Data Extract (ODE)		EXTRACT	* User exits for DEDB Unload or DEDB Reload
Managing Index	DEDB Index building			INDEXBLD	
Misc.	Area initialization			INITIALIZE	

New in 2014

New in V1.3

Enhancement in 2014

Task Efficiency

Area initialization function of FPA

SDEP relocation function of FPA UNLOAD and RELOAD

Checkpoint restart for FPA UNLOAD function

Usability improvements

Area Analysis Report enhancements

Area initialization function of FPA

- The **INITIALIZE** command provides concurrent initialization of multiple areas and MADS

- **DBRC=IGNORE** option can be used to ignore the DBRC requirement
  - No use of DBRC even if DBRC=FORCE is set for IMS

- **ODSNMASK** and **ALLOCATE** subcommands can be used to dynamically allocate spaces for the areas to initialize

- This function can provide better elapsed time and CPU time compared to DBFUMIN0 to initialize a large number of areas

An example JCL stream

```
//HFPINIT  EXEC PGM=HFPMAIN0
//STEPLIB  DD DISP=SHR,DSN=TOOLS.SHFPLMD0
//         DD DISP=SHR,DSN=IMS.SDFSRESL
//IMSACB   DD DISP=SHR,DSN=IMS.ACBLIB
//HFPRPTS  DD SYSOUT=*
//HFPPRINT DD SYSOUT=*
//HFPSYSIN DD *
  GLOBAL
    DBRC=IGNORE
  INITIALIZE
    DBD=DEDBJN22,
    OAREA=(DB22AR0,DB22AR1,DB22AR2),
    ODSNMASK='HPFP.&AREA.ADS1',
    OTASKCTL=2
  ALLOCATE
    OAREA=(DB22AR0,DB22AR1,DB22AR2),
    SPACE=(CYL,100),
    STORCLAS=STORCLAS,
    ATTRIBUTE=(DELETE)
/*
```

SDEP relocation function of FPA UNLOAD and RELOAD

- Using FPA Unload/Reload or Change function, you can enlarge or reduce the size of the SDEP part with the following restrictions:
  - The CI size must be the same in the old and new databases
  - The definition of SDEP segment must be the same in the old and new databases

- With **SDEP relocation function**, you can
  - Adjusts the SDEP logical begin (LB) to the first SDEP CI, and reduces/enlarges the size of SDEP part (Figure 1)
  - Resolves the wraparound state of the SDEP parts in the input area data set or image copy, and reduces or enlarges the size of SDEP part (Figure 2)
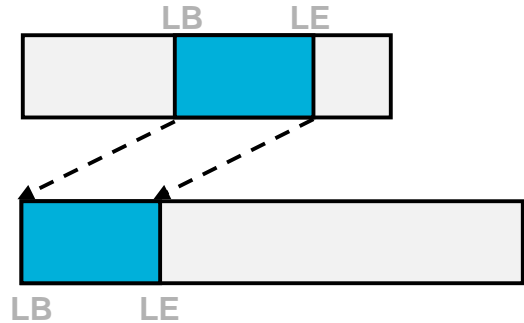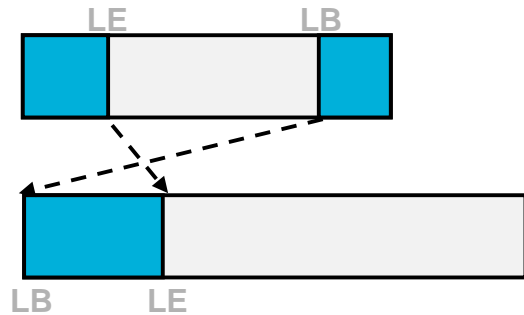
Figure 1

Figure 2

LB: Logical Begin
LE: Logical End

Checkpoint restart for FPA UNLOAD function

- FPA UNLOAD function provides checkpoint restart capability
  - Reprocess only the failed areas, without reprocessing previously successfully processed areas
  - Rerun the same JCL stream that was used in the previous run

- **RESTART=YES** and the checkpoint data set DD
  - These enable the checkpoint restart capability

- **RESTART_COUNT**
  - Specifies the maximum number of restart processing
    - The checkpoint data set will be cleared when the number of restart processing runs exceeds the specified number
    - Default: 5

- **Process Summary report**
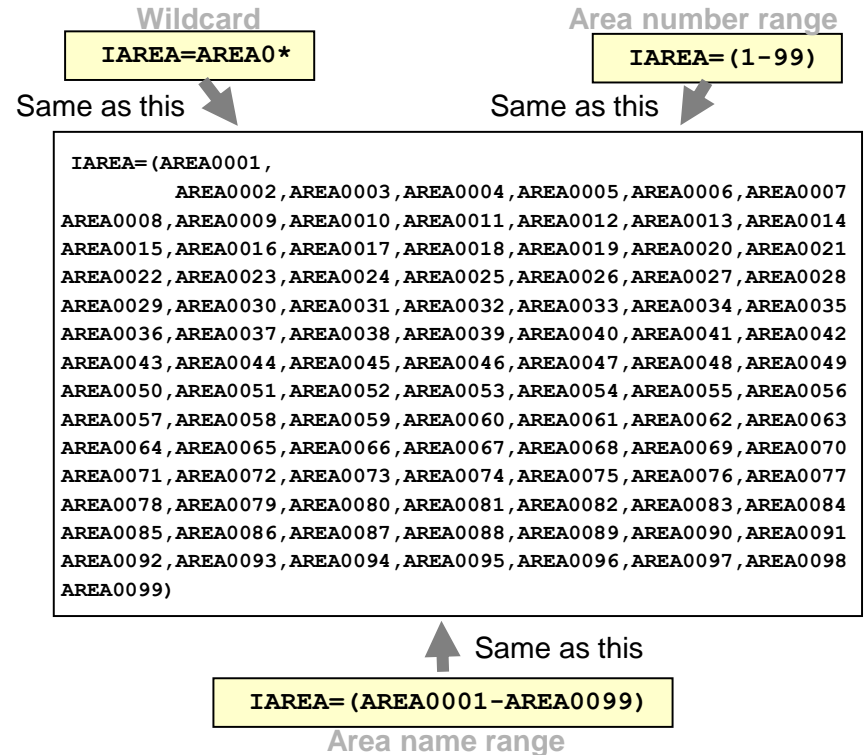 - Summarizes the processing result of each area

Usability improvements

- Simpler selection of areas (FPO and FPA)

- Timestamps in process messages (FPA)

- ADS name from RECON without authorization (FPA)

Simpler selection of areas (FPO and FPA)

- By a wildcard pattern
  - * represents 0 to 8 characters
  - % represents any single character

- By a range
  - A range of area numbers
  - A range of area names (in alphabetical order)

✓ This function is supported by the following utility commands:
  - All commands of FPA utility
  - All commands of FPO utility except for EXTEND (online area extend), which processes a single area at a time

Wildcard
```
IAREA=AREA0*
```

Area number range
```
IAREA=(1-99)
```

Same as this   Same as this

```
IAREA=(AREA0001,
        AREA0002,AREA0003,AREA0004,AREA0005,AREA0006,AREA0007
AREA0008,AREA0009,AREA0010,AREA0011,AREA0012,AREA0013,AREA0014
AREA0015,AREA0016,AREA0017,AREA0018,AREA0019,AREA0020,AREA0021
AREA0022,AREA0023,AREA0024,AREA0025,AREA0026,AREA0027,AREA0028
AREA0029,AREA0030,AREA0031,AREA0032,AREA0033,AREA0034,AREA0035
AREA0036,AREA0037,AREA0038,AREA0039,AREA0040,AREA0041,AREA0042
AREA0043,AREA0044,AREA0045,AREA0046,AREA0047,AREA0048,AREA0049
AREA0050,AREA0051,AREA0052,AREA0053,AREA0054,AREA0055,AREA0056
AREA0057,AREA0058,AREA0059,AREA0060,AREA0061,AREA0062,AREA0063
AREA0064,AREA0065,AREA0066,AREA0067,AREA0068,AREA0069,AREA0070
AREA0071,AREA0072,AREA0073,AREA0074,AREA0075,AREA0076,AREA0077
AREA0078,AREA0079,AREA0080,AREA0081,AREA0082,AREA0083,AREA0084
AREA0085,AREA0086,AREA0087,AREA0088,AREA0089,AREA0090,AREA0091
AREA0092,AREA0093,AREA0094,AREA0095,AREA0096,AREA0097,AREA0098
AREA0099)
```

Same as this

```
IAREA=(AREA0001-AREA0099)
```
Area name range

Timestamps in process messages (FPA)

- **MSTIMESTAMP=LOCAL**
  - Specifies that an event message be prefixed with the local timestamp of the event
  - Is supported for all FPA functions

MSTIMESTAMP=NO (default)

```
HFPA0001I ANALYZE PROCESSING STARTED FOR DATABASE: DEDBJN30, AREA NO: 1, AREANAME: DB30AR0
HFPD0001I A002SMSG DYNAMICALLY ALLOCATED TO
          - DSNAME: SYSOUT=*
HFPD0001I M002SMSG DYNAMICALLY ALLOCATED TO
          - DSNAME: SYSOUT=*
HFPA0002I NO POINTER ERRORS WERE DETECTED IN DATABASE: DEDBJN30, AREA NO: 1, AREANAME: DB30AR0
HFPA0004I ANALYZE PROCESSING COMPLETED FOR DATABASE: DEDBJN30, AREA NO: 1, AREANAME: DB30AR0
HFPL0001I REPORT STARTED FOR COMMAND: RELOAD, DATABASE: DEDBJN30
HFPL0002I REPORT COMPLETED FOR COMMAND: RELOAD, DATABASE: DEDBJN30
```

MSTIMESTAMP=LOCAL

```
2013.021 22:13:02:21 HFPA0001I ANALYZE PROCESSING STARTED FOR DATABASE: DEDBJN30, AREA NO: 1, AREANAME: DB30AR0
2013.021 22:13:02:28 HFPD0001I A002SMSG DYNAMICALLY ALLOCATED TO
                               - DSNAME: SYSOUT=*
2013.021 22:13:02:28 HFPD0001I M002SMSG DYNAMICALLY ALLOCATED TO
                               - DSNAME: SYSOUT=*
2013.021 22:13:02:52 HFPA0002I NO POINTER ERRORS WERE DETECTED IN DATABASE: DEDBJN30, AREA NO: 1, AREANAME: DB30AR0
2013.021 22:13:02:52 HFPA0004I ANALYZE PROCESSING COMPLETED FOR DATABASE: DEDBJN30, AREA NO: 1, AREANAME: DB30AR0
2013.021 22:13:02:52 HFPL0001I REPORT STARTED FOR COMMAND: RELOAD, DATABASE: DEDBJN30
2013.021 22:13:02:56 HFPL0002I REPORT COMPLETED FOR COMMAND: RELOAD, DATABASE: DEDBJN30
```

ADS name from RECON without authorization (FPA)

- **Issue**
  - Would like to schedule an FPA ANALYZE job to pickup area data set names from RECON without area authorization

- **Solution**
  - A new option **AREADSN=RECON** with DBRC=NO
    - FPA does sign-on/off to DBRC to obtain the area data set names from the RECON, but does not issue area authorization calls to DBRC
  - The AREADSN parameter is valid only for ANALYZE and TUNE functions

```
//HFP       EXEC PGM=HFPMAIN0
//STEPLIB  DD DISP=SHR,DSN=HPFP.SHFPLMD0
//         DD DISP=SHR,DSN=IMSVS.SDFSRESL
//IMSACB   DD DISP=SHR,DSN=IMSVS.ACBLIB
//RECON1   DD DISP=SHR,DSN=IMSVS.RECON1
//RECON2   DD DISP=SHR,DSN=IMSVS.RECON2
//RECON3   DD DISP=SHR,DSN=IMSVS.RECON3
//HFPRPTS  DD SYSOUT=*
//HFPPRINT DD SYSOUT=*
//HFPSYSIN DD *
  GLOBAL
     DBRC=NO,
     AREADSN=RECON
  ANALYZE
     DBD=DEDBJN22,
     IAREA=ALL,
     PTRCHKLVL=FULL
/*
```

Another option of AREADSN

- **AREADSN=NORECON** (specified with DBRC=NO) suppresses access to RECON
- With this option, FPA obtains area data set names from *areaname* DD statements or from the utility control statements.

Area Analysis Report enhancements

- **DEDB Area Analysis report** *(enhanced in FP Solution Pack V1.2)*
  - New report header lines that indicate:
    - Input data set name
    - Input data set type (and the timestamp for image copy)
  - SDEP information in the following reports:
    - SDEP LB/LE information in Free space Analysis report
    - Number of occurrences in Segment Placement Analysis report
  - Segment prefix length information in the DB Record Profile Analysis report

- **DB Record Length Distribution report** *(2014 enhancement for V1.3)*

- **Segment Length Distribution report** *(introduced in V1.2)*

- **Consolidated Largest Database Records report** *(introduced in V1.2)*

Tuning Aid Function

FPA TUNE function (introduced in FP Solution Pack V1.3)

▪ FPA TUNE simulates changes in the DBD parameters and randomizers without loading the segments

▪ It helps you select the physical database attributes that satisfy your performance and space utilization requirements

▪ Changes in any or all of the following database specifications can be evaluated by the segment load simulation:
  – Randomizer
  – Number of database areas
  – UOW parameter values for each area
  – ROOT parameter values for each area
  – CISIZE for each area

A sample JCL stream for FPA TUNE

```
//FPATUNE EXEC PGM=HFPMAIN0
//STEPLIB DD DISP=SHR,DSN=HFP.SHFPLMD0
//        DD DISP=SHR,DSN=HPIC.SHPSLMD0
//IMSACB   DD DISP=SHR,DSN=user.ACBLIB
//IMSDALIB DD DISP=SHR,DSN=user.MDALIB
//HFPSYSIN DD *
  GLOBAL DBRC=NO
  TUNE
    DBD=DEDBJN30,
    IAREA=DB30AR0,
    TARDSNMASK='HPFP.DB30AR0.TAR',
    AREASCAN=YES
  SIMULATE
    OAREA=DB30AR0,
    RMODLIB='USER.RMODLIB',
    RMOD=TSTRAND1
  AREADEF
    OAREA=DB30AR0
  THRESHOLD
    DOVFFS_PCT=30,
    IOVFFS_PCT=50
/*
```

FPA TUNE with offline areas or image copies

- The Tune function scans the database and simulates the tuned areas using input area and tuning definitions

- The Tune function generates reports of the tuning summary and area analysis for current areas and simulated areas.

- The Tune function also generates **Tuning Aid records (TARECs)**, that enable to run the Tune function iteratively without scanning the database.

1st Trial

Area or IC

Tuning definitions

FPA TUNE

Reports
- Tuning summary
- Area analysis

- ACBLIB
- (optional) tuned ACBLIB
- Control statements

Tuning Aid records

Subsequent Trials
(can be run iteratively)

FPA TUNE

Tuning definitions

Reports
- Tuning summary
- Area analysis
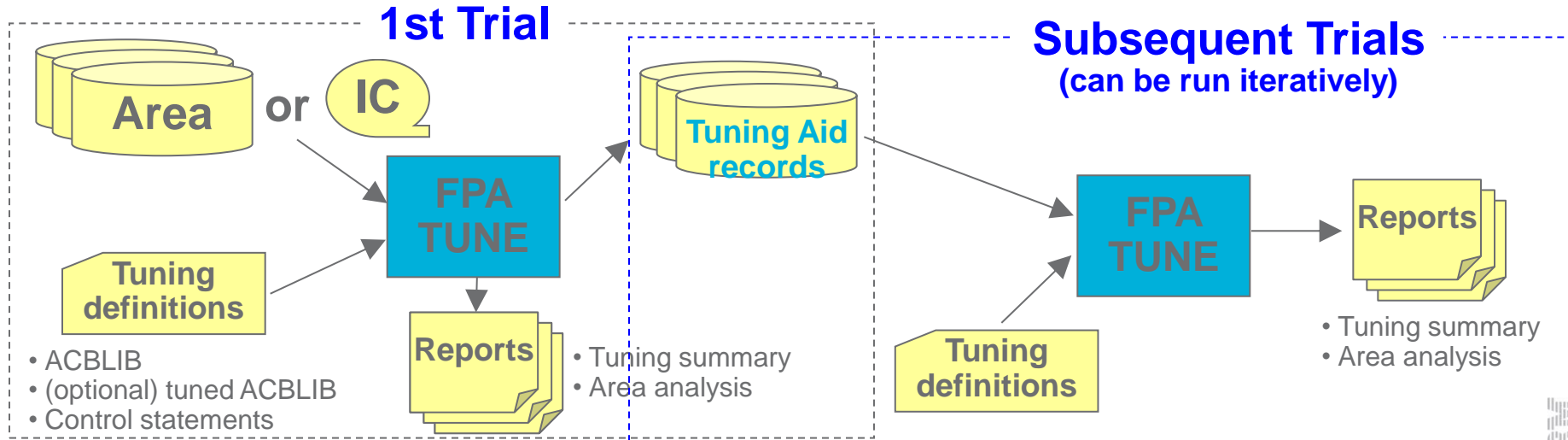
FPA TUNE using online areas as input

- By using OPC, you can simulate DEDB tuning without making areas offline

- OPC with TARECGEN=YES or TARECGEN=ONLY can be used to scan the online areas and generates the Tuning Aid records (TARECs)

- Use TARECGEN=YES if you want to see the tuning result immediately in the OPC job

- Use TARECGEN=ONLY to suppresses the tuning simulation in the OPC job, which is recommended if you want to reduce CPU time and shorten the elapsed time of the OPC job that runs in the same LPAR as an online IMS system does
  - The FPA TUNE function in another job (or in another jobstep) can simulate tunings and generate reports with multiple different DBD parameters using the TARECs
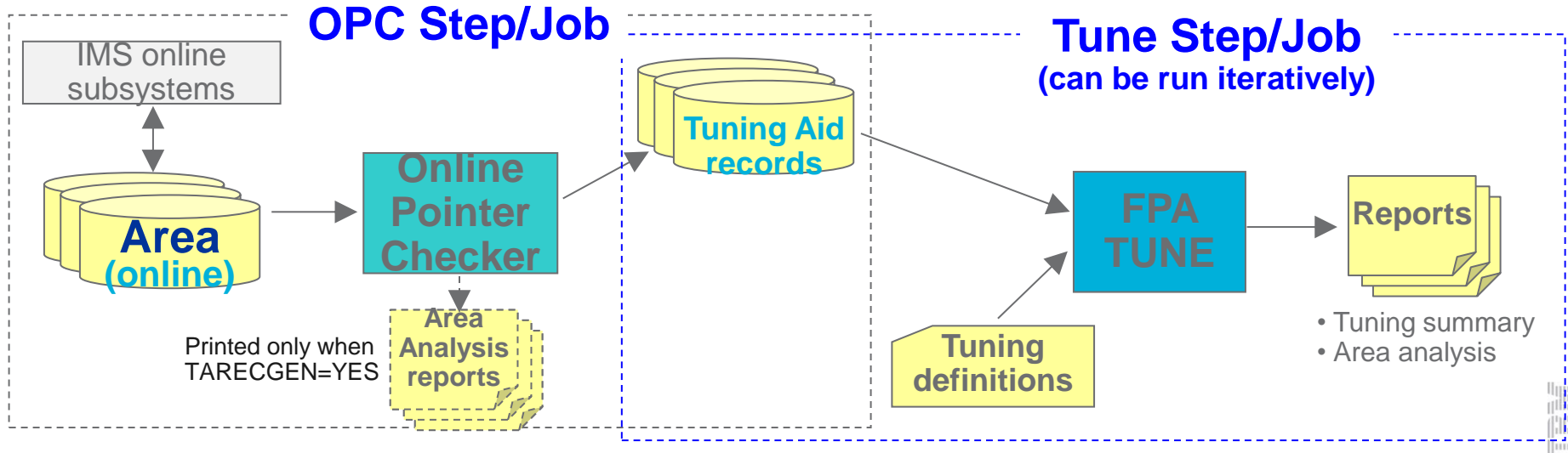
Improvement from FPB DEDB Tuning Aid
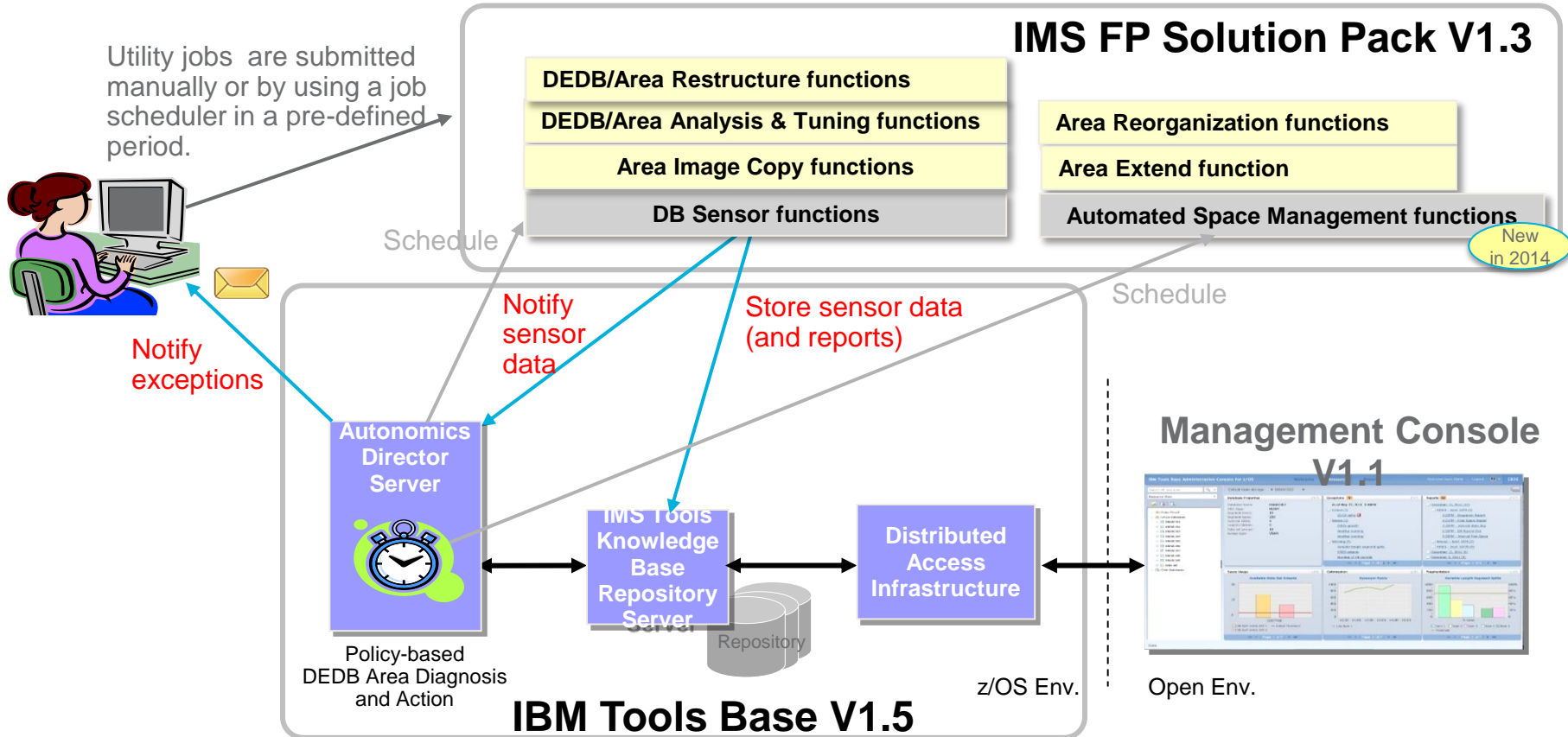
- **Usability improvement**
  - The number of job steps has been reduced
    - For an offline database, from 12 steps to 1 step.
    - For an online database, from 11 steps to 2 steps.
  - A temporary ACB library that contains a new area definition is not required
    - Area definition parameters to tune can be specified by utility control statements
    - Optionally, tuned ACB library can be used as input.
  - Effect of database tuning can be evaluated easier than before
    - New **Area Tuning Summary Report** is provided

- **Performance improvement**
  - Elapsed time and CPU time of FPA Tune are reduced from those of DEDB Tuning Aid

Enhanced capabilities with IBM Tools Base

Utility jobs are submitted manually or by using a job scheduler in a pre-defined period.

IMS FP Solution Pack V1.3

DEDB/Area Restructure functions

DEDB/Area Analysis & Tuning functions

Area Image Copy functions

DB Sensor functions

Area Reorganization functions

Area Extend function

Automated Space Management functions

New in 2014

Schedule

Schedule

Notify sensor data

Store sensor data (and reports)

Notify exceptions

Autonomics Director Server

Policy-based DEDB Area Diagnosis and Action

IMS Tools Knowledge Base Repository Server

Repository

Distributed Access Infrastructure

Management Console V1.1

z/OS Env.

Open Env.

IBM Tools Base V1.5

Autonomics functions

Database Sensor function

Online Space Management utility

FP Active Autonomics

Database Sensor function

- Database Sensor collects and stores DEDB area statistics data in the Sensor Data repository of IMS Tools Knowledge Base

- The stored data can be used by Policy Services, Autonomics Director, and Management Console for database analysis and tuning purposes

- Database Sensor function (**SENSOR=YES** option) is provided by the following tools:
    - **Integrated DB Sensor**, which runs inside the following utilities
        - Fast Path Advance Tool (FPA)
        - Fast Path Online Tools (FPO)
        - IMS High Performance Image Copy (HPIC)
    - **Stand-alone DB Sensor**
        - Fast Path Stand-alone DB Sensor (FP Stand-alone DB Sensor)

- Sensor data can be printed by the following tool:
    - **Fast Path DB Sensor Printing utility**

Sensor data for DEDB area

- These categories of sensor data can be collected and stored for a DEDB area

- A complete list of the data elements supported by the DB Sensor can be found in [the Reference section of the *IBM Tools Base Policy Services User's Guide*](.).

- **Area level statistics**
  - Area definition
  - Free space percentages
    - RAA Base, DOVF, IOVF, and SDEP
  - Usage of overflow (DOVF and IOVF)
  - Number of segment occurrences
  - Database record stats
  - Randomizing synonym stats
  - Physical I/O stats
  * Last three are optional, but recommended

- **UOW or UOW Group level statistics (optional)**
  - Free space percentages
  - Overflow usage
  - Number of segment occurrences
  - Database record stats
  - Randomizing synonym stats
  - Physical I/O stats

FP integrated DB Sensor – Sensor function running inside utilities

FPA commands that support the integrated DB Sensor

```
UNLOAD
O RELOAD
O CHANGE
O REORG
O ANALYZE
TUNE
EXTRACT
DMACPRT
INDEXBLD
INITIALIZE
```
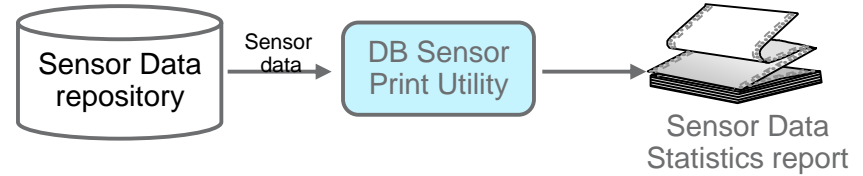
A Sample JCL Stream for FPA

```
//HFP       EXEC PGM=HFPMAIN0
//STEPLIB  DD DISP=SHR,DSN=HPFP.SHFPLMD0
//         DD DISP=SHR,DSN=IMSVS.SDFSRESL
//         DD DISP=SHR,DSN=ITB.SHKTLOAD
//IMSACB   DD DISP=SHR,DSN=IMSVS.ACBLIB
//IMSDALIB DD DISP=SHR,DSN=IMSVS.MDALIB
//HFPRPTS  DD SYSOUT=*
//HFPPRINT DD SYSOUT=*
//HFPSRPT  DD SYSOUT=*
//HFPSYSIN DD *
  GLOBAL  DBRC=YES,
          ITKBSRVR=FPQSRV01,
          ADXCFGRP=ADSRV01
  ANALYZE DBD=DEDBJN22,
          IAREA=(DB22AR0),
          PTRCHKLVL=FULL,
          SENSOR=YES,
          SENSOR_DBREC=YES

  REPORT
/*
```

FPO utilities (commands) that support the integrated DB Sensor

```
O OPC (ANALYZE)
  ODE
  OAE
O OER (REORG with PTRCHK=YES)
  ODM
```

A Sample JCL Stream for FPO

```
//HFP     EXEC FPUTIL,DBD=DEDBJN23,IMSID=IMS1
//FPU.STEPLIB DD DISP=SHR,DSN=HPFP.SHFPLMD0
//            DD DISP=SHR,DSN=IMSVS.SDFSRESL
//            DD DISP=SHR,DSN=ITB.SHKTLOAD
//FPXMFILE    DD SYSOUT=*
//FPXRPTS     DD SYSOUT=*
//FPXSPRT     DD SYSOUT=*
//FPXIN DD *
  GLOBAL   DBRC=YES,
           ITKBSRVR=FPQSVR01,
           ADXCFGRP=ADSVR01
  ANALYZE AREA=ALL,
          TYPRUN=RPT,
          FULLSTEP=YES,
          SENSOR=YES,
          SENSOR_DBREC=YES
/*
```

- The **ADXCFGRP** option is needed so that Tools Base Autonomics Director can be notified of a new sensor data for evaluation
- **SENSOR_DBREC=YES** is needed to store database record stats, randomizing synonym stats, and physical I/O stats
- These options can be specified as a site default option

FP Stand-alone DB Sensor

- Can processes both offline and online DEDB areas

- Can processes multiple areas of a DEDB in a job

- FP Stand-alone DB Sensor does not verify segment pointer integrity

- If you want to collect and store sensor data only when database integrity is verified, use an FPA or FPO, or HPIC job with SENSOR=YES option specified

- Considerations on online area processing
  - If an FP Stand-alone DB Sensor runs against an online area, the data element values might be collected while the area is being updated
  - For sensor data integrity, use either of the following methods:
    - Quiesce the area or take the area offline before running the Sensor job
    - Use OPC's DB Sensor function

A Sample JCL Stream

```
//HFP       EXEC PGM=HFPSENSR
//STEPLIB  DD DISP=SHR,DSN=HPFP.SHFPLMD0
//         DD DISP=SHR,DSN=IMSVS.SDFSRESL
//         DD DISP=SHR,DSN=IMSVS.PGMLIB
//         DD DISP=SHR,DSN=ITB.SHKTLOAD
//IMSACB   DD DISP=SHR,DSN=IMSVS.ACBLIB
//IMSDALIB DD DISP=SHR,DSN=IMSVS.MDALIB1
//HFPSPRT  DD SYSOUT=*
//HFPPRINT DD SYSOUT=*
//HFPSYSIN DD *
  GLOBAL   DBRC=YES,
           ITKBSRVR=FPQSRV01,
           ADXCFGRP=ADSRV01
  DATABASE DBD=DEDBJN23,
           IAREA=*ALL
/*
```

FP DB Sensor Printing Utility

- DB Sensor Printing utility
  - Reads the <u>latest</u> sensor data for the specified DEDB area from the IMS Tools KB Sensor Data repository, and print a Sensor Data Statistics report in the HFPSPRT data set
  - Can print the sensor data that is related to UOW in the Sensor Data Statistics report when the UOW_REPORT=YES is specified

- Current restrictions
  - Only one area can be processed in a job step
  - Only the latest sensor data for an area can be printed

Sensor Data repository → Sensor data → DB Sensor Print Utility → Sensor Data Statistics report

A Sample JCL Stream

```
//HFPSPRNT EXEC PGM=HFPSPRNT
//STEPLIB  DD DISP=SHR,DSN=HPFP.SHFPLMD0
//         DD DISP=SHR,DSN=ITB.SHKTLOAD
//RECON1 DD DISP=SHR,DSN=HPFP.RECON1
//HFPSPRT DD SYSOUT=*
//HFPSOUT DD SYSOUT=*
//HFPSPRIN DD *
  GLOBAL    ITKBSRVR=FPQSVR01
  DATABASE DBD=DEDBJN23,
           AREA=DB23AR1
/*
```

Requirement: Area space management automation

- Provide a new Policy Services rules and policy for Fast Path DEDB to enable "active" actions for the following recommendations for a DEDB area:
  - REORG is needed
  - EXTEND is needed

- Use the policy to determine when a DEDB area needs online reorganization or online area extender to be run and actually initiate the run

Free spaces and segments in UOWs and IOVF are reorganized and relocated within the same data set.

REORG

A DEDB Area Data Set

UOWs	IOVF Part	SDEP Part

EXTEND IOVF   EXTEND SDEP

A new area data set is allocated in a new space and segments are copied to the new ADS with the size of IOVF part and/or SDEP part extended to avoid space run-out in these portions of the area.
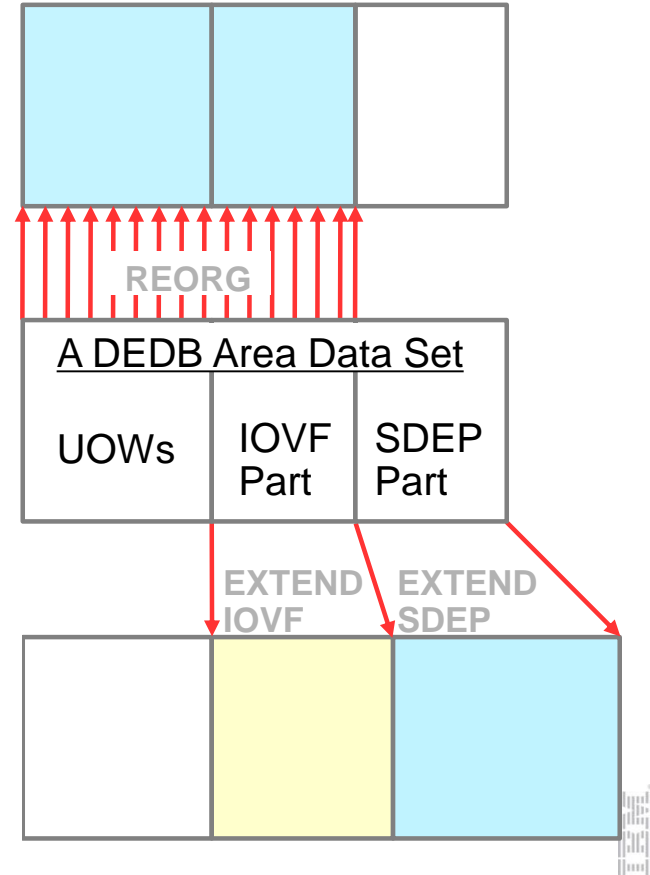
Our solution: FP Active Autonomics

▪ An area is reorganized or expanded only when Autonomics Director (AD) detects exceptions by automated sensor data evaluation

▪ (Possibly) multiple area space management actions can be taken in one OSM job step automatically based on the exceptions and associated action recommendations by AD
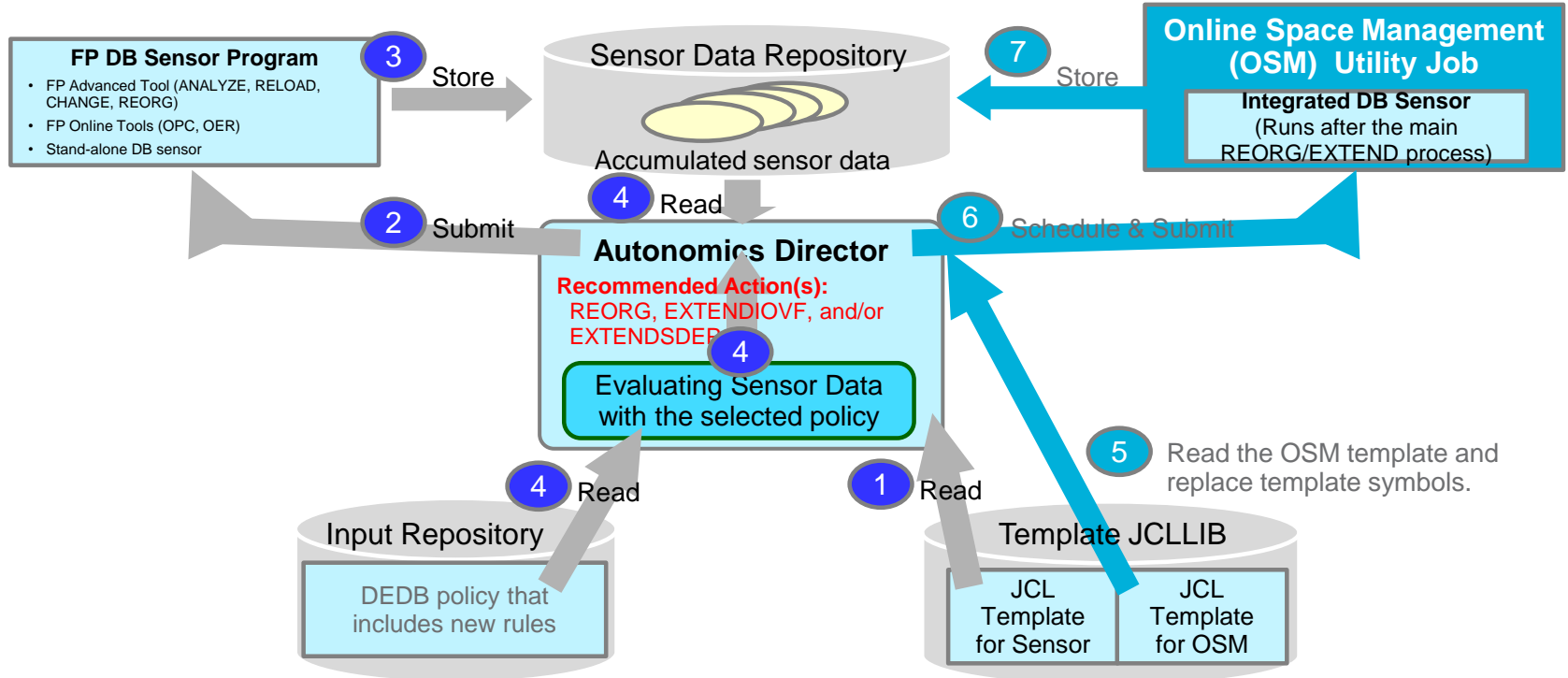
Policy Services rules used to evaluate sensor data

- Sensor data evaluation for an area is done by using IBM Tools Base Policy Services

- Policy Services runs automatically non-peak hours, which is defined in AD, and based on an AD Monitor List definition for the area

- The Monitor List definition includes the evaluation policy to be used and the timing of evaluation
  - The following sensor data evaluation rules can be used:
    - **IBM.DEDB_RFS.11** and **IBM.DEDB_RFS.21**
      … used to detect a need for free space reorganization
    - **IBM.DEDB_FS.31**
      … used to detect a need for IOVF expansion
    - **IBM.DEDB_FS.81**
      … used to detect a need for SDEP expansion

- A policy evaluation can detect exceptional states of the area space usage and can recommend REORG, EXTENDIOVF, or EXTENDSDEP action to be taken by OSM for the area

- AD can be configured to schedule an OSM job in a database maintenance window when a REORG or EXTENDxxxx action is recommended

Online Space Management utility (OSM)

- OSM integrates the following FPO functions for a DEDB area into one job step:
  - **Online free space reorganization**
    - Reorganizing a set of UOWs in the area based on the ranked set of UOWs
  - **Online area expansion**
    - Extending IOVF portion, SDEP portion, or both of the area
  - **Online pointer checking and sensor data collection**
    - Verify the DEDB integrity, generating Area Analysis reports, and collect and store sensor data for the area after the area is reorganized or expanded

- These functions are provided internally by the following utility functions:
  - Reorganization – Online Expert Reorganization (OER)
  - Area expansion – Online Area Extender (OAE)
  - Pointer checking and sensor data collection – Online Pointer Checker (OPC)

SDEP_CI and IOVF_UOW in percentage

- The extended size for SDEP portion can be specified with a specific number or by a percentage of CIs within the total number of CIs defined in the SDEP portion
  - SDEP_CI=xx or xx%

- The extended size for IOVF portion can be specified with a specific number of by a percentage of UOWs within the total number of UOWs defined in the IOVF portion
  - IOVF_UOW=xx or xx%

- These specifications allow OAE to be executed multiple times without JCL modification

An example scenario of FP Active Autonomics (1/3)

1. Everyday after 8:00 p.m., Autonomics Director initiates an OPC utility job for each of the 5 areas. The OPC utility stores the sensor data and notifies Autonomics Director.

2. Autonomics Director evaluates sensor data that is collected from each area, and detected the following conditions:

An example scenario of FP Active Autonomics (2/3)

3. On Sunday after 3:00 a.m., during the defined database maintenance window, Autonomics Director submits the OSM utility job for each area to which some actions were recommended by the last policy evaluation.

A case where all OSM jobs succeeded

```
 Menu   View   Help
 _____
                    Autonomics Director Reorg Job Status      Row 1 to 2 of 2

 Locale  . . :  RECON6AC         Group type  :  USERGRP
 Group name . . . . . :   FP6AC
 Group description . . :   FOR TEST
 Reorg-Date  . . . . . :   Jun 05,'14        Reorg-Time  . . . . . . :   15:33:43


 Row Actions:    S - View the job details

 Act    Job type   Job name   Job num   Job status        End status   Completed
 __     REORG      DBAR001R   JOB01024  Ended             RC00         Successful
        REORG      DBAR002R   JOB01025  Ended             RC00         Successful
 ********************************* Bottom of data *******************************
```

> DBAR001 and DBAR002 are expanded or reorganized successfully

A case where an OSM job failed

```
 Row Actions:    S - View the job details

 Act    Job type   Job name   Job num   Job status        End status   Completed
 __     REORG      DBAR001R   JOB01001  Ended             RC08(OAE)    Failed
        REORG      DBAR002R   JOB01002  Ended             RC00         Successful
 ********************************* Bottom of data *******************************
```

> The OSM job for DBAR001 failed in OAE process, and the OSM job for DBAR002 succeeded

An example scenario of FP Active Autonomics (3/3)

4. Autonomics Director evaluates the sensor data that was notified by the OSM utility job. As result of policy evaluation, Autonomics Director did not detect exceptional state in the two reorganized or expanded areas. Autonomics Director recommends no action for those areas.

The exceptional states of DBAR001 and DBAR002 have been resolved after OSM jobs completed

Simplifying access
to monitoring information

Introducing the new…
IBM Management Console
for IMS and DB2 for z/OS 1.1

IBM Management Console
for IMS and DB2 for z/OS

- Provides a single, holistic easy-to-use interface to manage IMS and DB2 systems and databases
  - Zero-install web-based interface
  - Consolidate information from IMS, DB2 and tools to paint a more complete picture from across the entire enterprise
  - Reduced time for problem identification and resolution through tight integration with IMS and DB2 Autonomics
  - Dramatically reduced learning curve for new users of IMS and DB2

- Now available as a separately orderable *no-charge* product *(5655-TAC)*
  - Extensible by growing number of products and solution packs adding value beyond the base

- Direct transparent upgrade from IBM Tools Base Admin Console 1.4 to Management Console 1.1

IBM®

IBM® Management Console for IMS™ and DB2® for z/OS® V1.1

User ID

Password

Help

Log In

Licensed Materials - Property of IBM. 5655-TAC © Copyright 2011, 2014 IBM Corp. IBM, the IBM logo, IMS, DB2, and z/OS are trademarks of IBM, the IBM logo, IMS, DB2, and z/OS are trademarks of IBM Corporation, registered in many jurisdictions worldwide. Other product or service names might be trademarks of IBM or other companies. This Program is licensed under the terms of the license agreement accompanying the Program. This license agreement may be either located in a Program directory folder or library identified as License or Non_IBM_License, if applicable, or provided as a printed license agreement. Please read this agreement carefully before using the Program. By using the Program, you agree to these terms.

IBM Management Console for IMS and DB2 for z/OS

- Drill down from the IMSPLEX level through:
  - IMSPLEX members
  - IMS Connect
  - IMS Subsystems
  - Transactions
  - Programs
  - Routing Code
  - Databases

- Identify, stop, start IMS resources

- Explore resource relationships

- OM Command Builder

IBM Management Console for IMS and DB2 for z/OS

- Graphical control of IMS Autonomics Director
  - Define monitor lists, setup peak times and maintenance windows

- Web access to IMS reports from a variety of Tools
  - Search and filter by type, date, completion code, etc
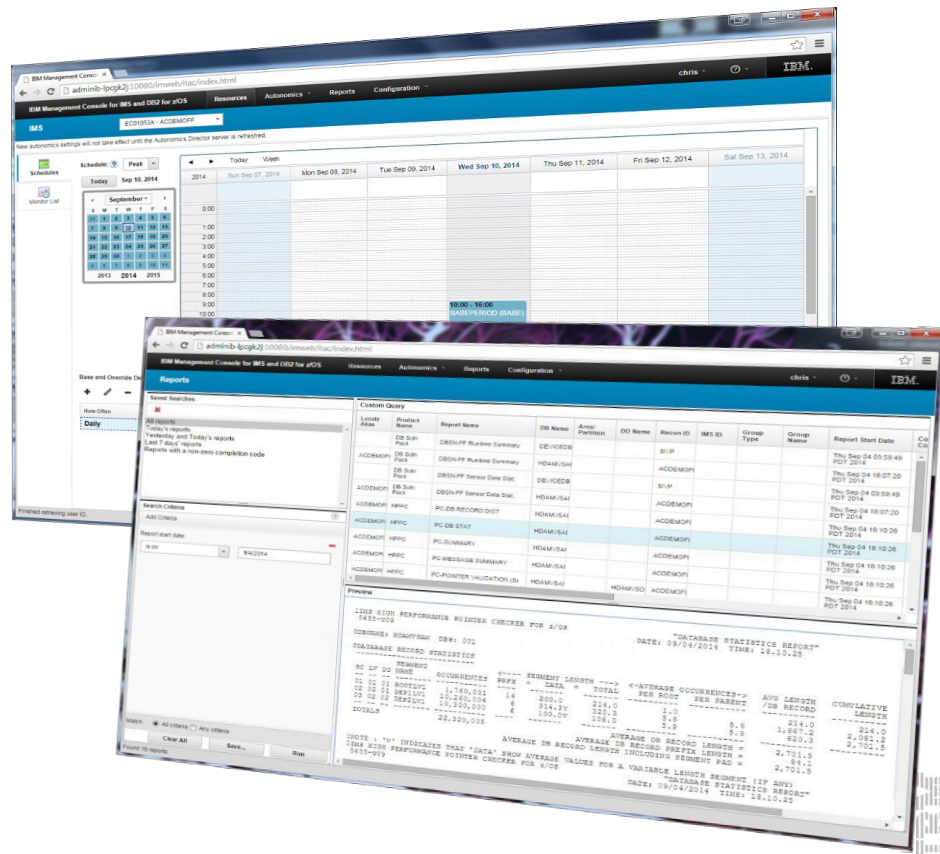
IBM Management Console for IMS and DB2 for z/OS

Extended with the IMS FF/FP Solution Pack

- Extended IMS Autonomics and Database information

- Easily identify and diagnose symptoms and recommended actions through IMS Database Autonomics

- Identify trends and make projections through database sensor charts

- Visualize IMS FF/FP Databases through **Library Integrity Utilities**
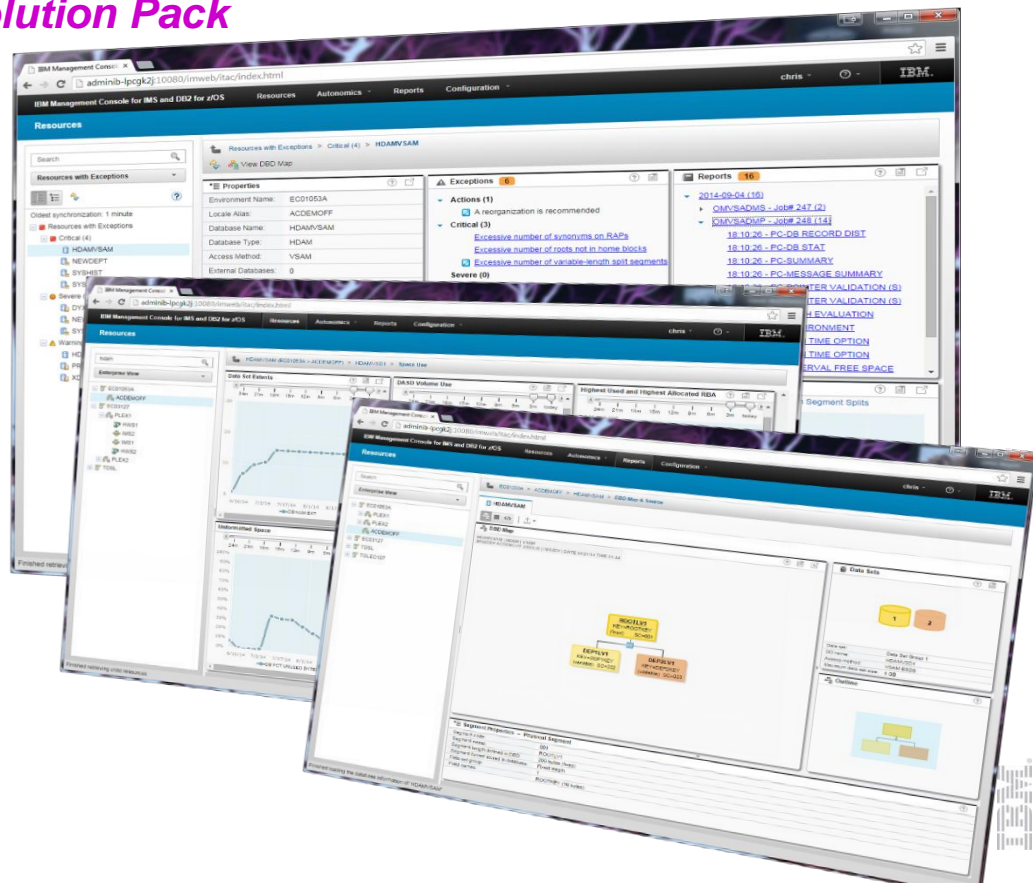
IBM Management Console for IMS and DB2 for z/OS

- Progressive drill down through variety of DB2 object dashboards

- **Autonomics Director for DB2 for z/OS** (in the *no-charge* Tools Base) enables:
  - Charting of DB2 object statistics through RTS snapshots with
  - Autonomics control to define profiles and maintenance windows
  - Integrated support for the DB2 Admin Task Scheduler
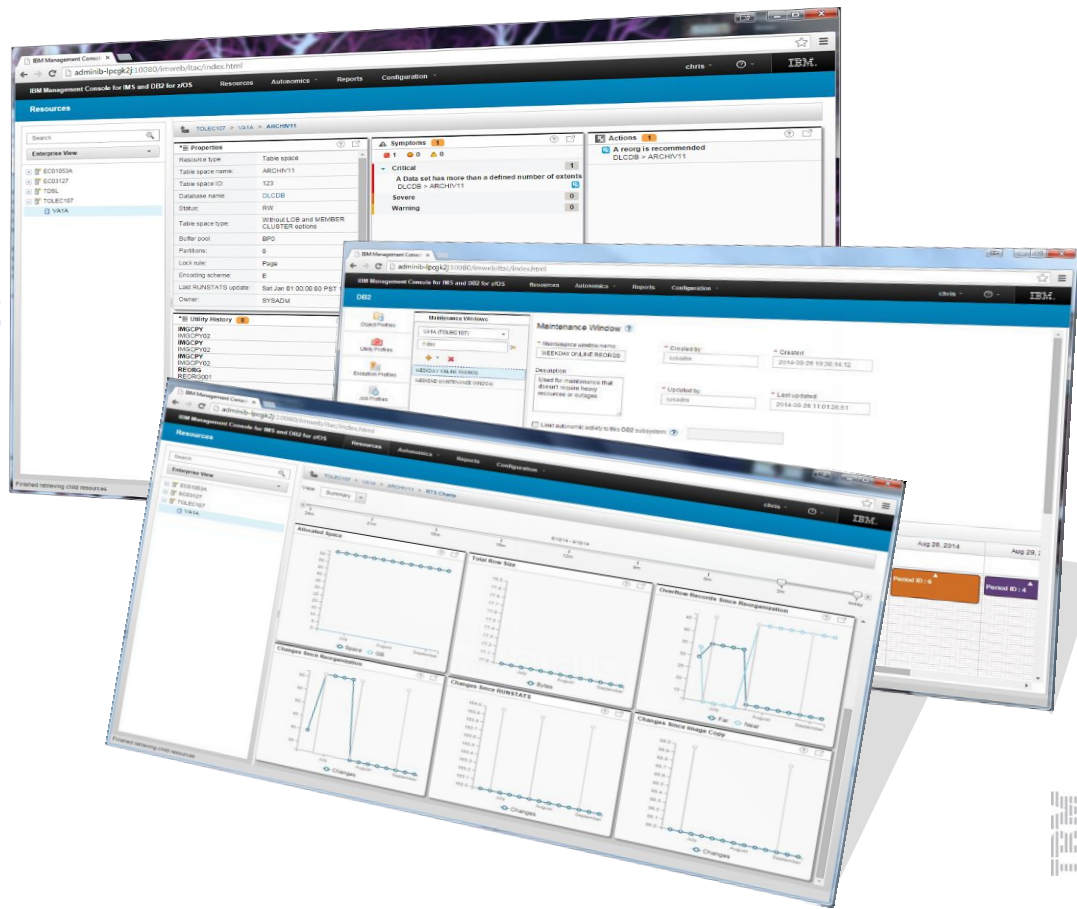
IBM Management Console for IMS and DB2 for z/OS

Extended with the DB2 Utility Solution Pack

- Identification and Diagnosis of symptoms and recommended actions for REORGs, ICs, Runstats

- Reporting on historical utility execution including timestamp, elapse time, system output, etc

- Graphical interfaces to define Automation Tool Object, Utility, Exception, and Job Profiles
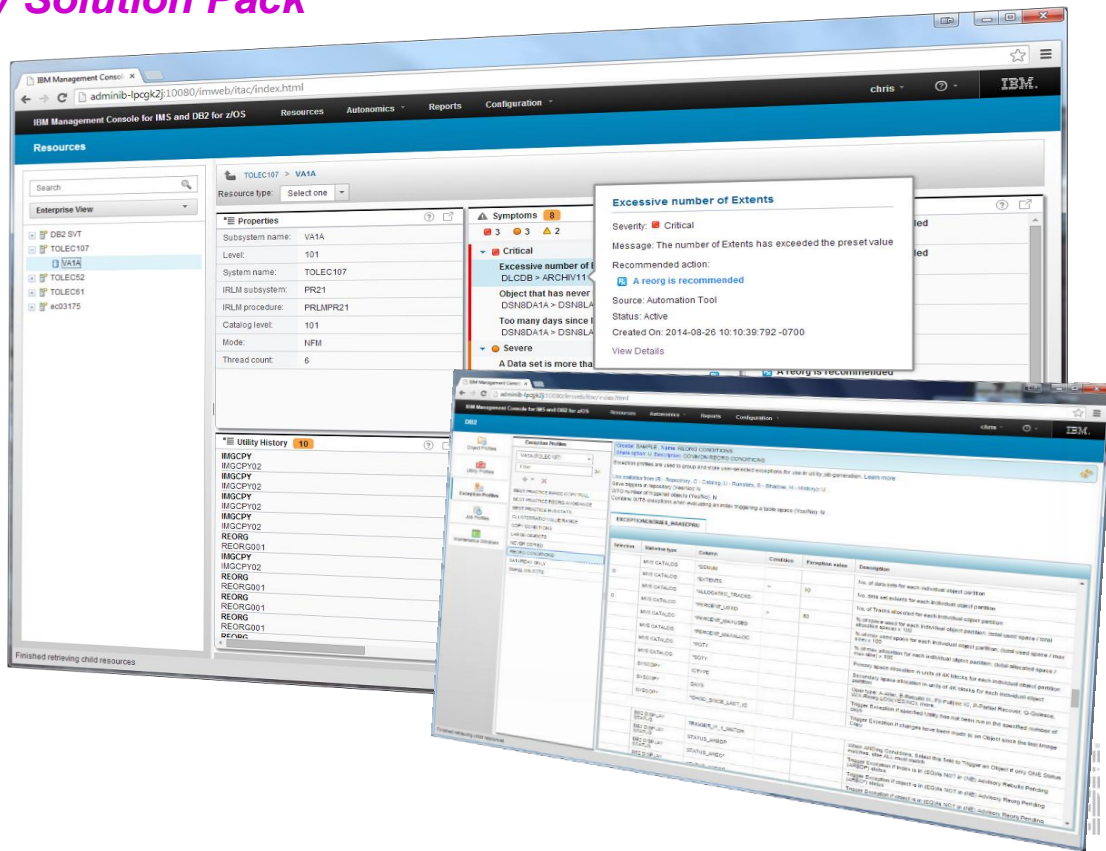
IBM Management Console for IMS and DB2 for z/OS

- Quickly identify and drill down to databases and objects that need your attention from a single starting point

- Easily manage by exception and recommendation, taking action before problems occur

- Rapidly interpret statistical trends to verify and project

- Shorten the learning curve for new administrators

- …all from a **unified IMS and DB2** interface
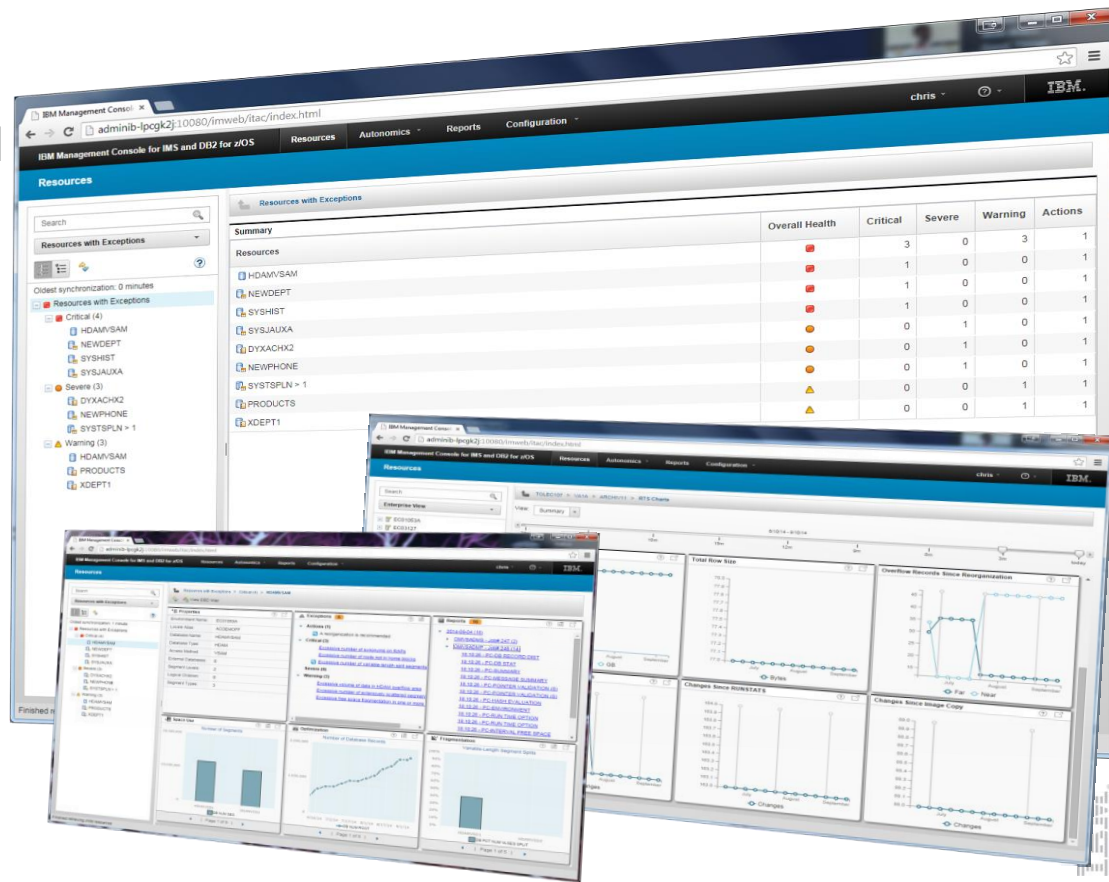
More sessions:

Tuesday 11:30
   B04: The IBM Management Console:
   ….Gain Insight into your Enterprise

Tuesday and Wednesday 16:00

   D03 (Hands-on-Lab) The IBM Management Console
   for IMS and DB2 for z/OS

Summary

- IMS Tools from IBM provide, integrated, easy-to-use solutions that fit your company's needs

- We are continuing to invest in our IMS Tools technology and have a vision for our IMS Tools that centers around autonomic computing

- IBM is dedicated to the continued success and support of IMS and the mainframe. We're invested for the long term, right beside you.

IMS Tools

APAR Information

APAR/PTF numbers for FP Solution Pack V1.3 new functions (1)

- **FPA TUNE function's must-have APAR/PTF**
  - PI06424/UI14684: Enhancement for Area Tuning Summary report and TARECGEN

- **FP Active Autonomics**
  - Fast Path Solution Pack V1.3 - PI06716/UI17205
  - IBM Tools Base V1.4 - PI09140 and PI09270/UI16631
  - IBM Tools Base V1.5 - FP Active Autonomics function is included in the base level

- **FPA INDEXBLD function enhancements for FP Secondary Index**
  - PI17756/UI23683: Online RESYNC for FP Secondary Index
  - PI17756/UI23683: Better handling of duplicate keys (DUPKEYMAX and DUPKEYACT)

APAR/PTF numbers for FP Solution Pack V1.3 new functions (2)

- **IMS V14 support**
  - PI22667/UI23956: This also includes the support for 64-bit buffers in OER

- **Other new functions**
  - PI16431/UI18578: Retrieving ADS name from RECON without authorization (AREADSN)
  - PI14356/UI18137: DB Record Length Distribution report (DBRLDIST)
  - PI16740/UI19724: OAE multi volume support (MAXVOLS and VOLUME_EXT)
  - PI18225/UI20175: Duplicate keyword check enhancement (KEYWORD_DUPCHK)
  - PI18788/UI19156: Setting the "empty flag" in 2nd CI when an area is initialized (to conform to the IMS V12/V13 enhancement)
  - PI24263/UI20801: Providing a sample JCL for stand-alone DB Sensor

FP Secondary Index Online Resync

Functions for Fast Path Secondary Index

- Brief overview of IMS FP Secondary Index (FPSI)

- FPA INDEXBLD function

- FPSI Online Resynchronization

Brief overview of IMS FP Secondary Index

- IMS FP Secondary Index support
  - Provides secondary index capability for DEDB
    - similar to that for full-function database (not exactly same)
  - Has capabilities that are not available with secondary indexes for full-function databases
    - such as user data partitioning and multiple secondary index segments

- More application capability support in IMS 13 (SSA Boolean operators and command codes)

Target segment
Source segment
Pointer segment
(always symbolic)
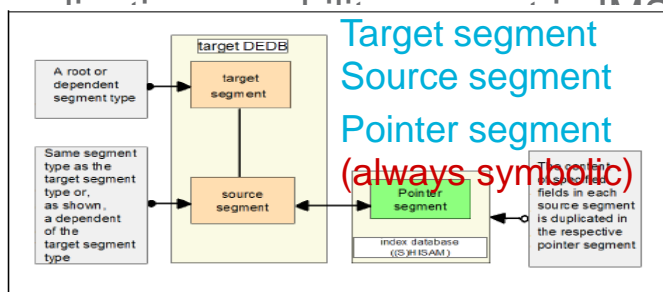
Figure 3-4 Target, source, and pointer segment

Non-unique keys with overflow or unique keys with various options (SUBSEQ=, DDATA=, /CK, …)
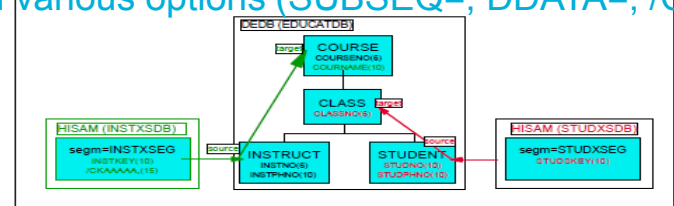
Figure 3-8 Secondary indexes on DEDB

User partitioning using a user partition selection exit routine
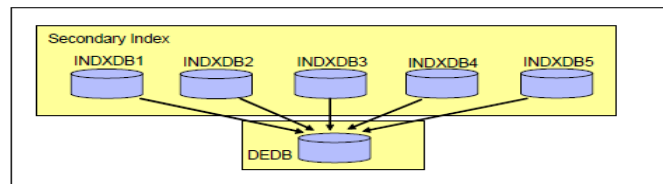
Figure 3-16 User partitioning for secondary indexes

Multiple secondary index segments
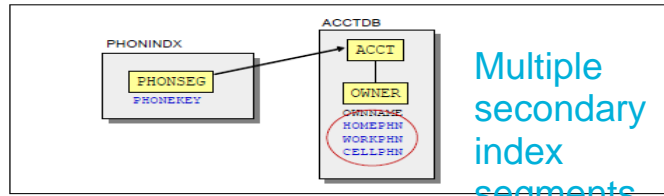
Figure 3-17 Multiple secondary index segments

(Source: All figures are cited from the Redbook *"IBM IMS Version 12 Technical Overview"*)

Index (re-)building solutions provided by FP Solution Pack

- FP Advanced (FPA) tool provides the capability of
  1. Building the secondary indexes (the **INDEXBLD** function)
     - Multiple secondary indexes in one job step with higher performance than loading segments into the target DEDB by using an IMS application program
  2. Verifying integrity of index pointer segments in secondary indexes (the **INDEXDBD** option of the ANALYZE function)
  3. Supporting the **IB(BLD_SECONDARY(…))** capability of DRF for the recovery of indexed DEDB areas
     - FP secondary indexes can be rebuilt during DEDB area recovery

Customer's Business Benefits
- Improved productivity by high performance index building/rebuilding
- Protection for loss of business from corrupted or inconsistent indexes
- Faster recovery from a failure in an indexed DEDB

Building multiple secondary indexes at a time

- FPA **INDEXBLD** function
  - Can build all secondary indexes for a DEDB when the secondary indexes are defined against the existing DEDB (**INDEXDBD=ALL**)
  - Can build only the added or broken secondary indexes

A Sample JCL Stream for FPA INDEXBLD

```
//FPA      EXEC PGM=HFPMAIN0
//STEPLIB  DD DISP=SHR,DSN=FPSP.SHFPLMD0
//         DD DISP=SHR,DSN=IMSVS.SDFSRESL
//         DD DISP=SHR,DSN=IMSVS.PGMLIB
//IMSACB   DD DISP=SHR,DSN=IMSVS.ACBLIB
//IMSDALIB DD DISP=SHR,DSN=IMSVS.MDALIB
//IMS      DD DISP=SHR,DSN=IMSVS.DBDLIB
//HFPSYSIN DD *
  GLOBAL DBRC=YES
  INDEXBLD DBD=DEDBJN22,ITASKCTL=4,
  IAREA=[ALL|(area1,area2,...)],
  INDEXDBD=[ALL|(indexdbd1,indexdbd2,…)]
/*
```

Build all indexes for a DEDB
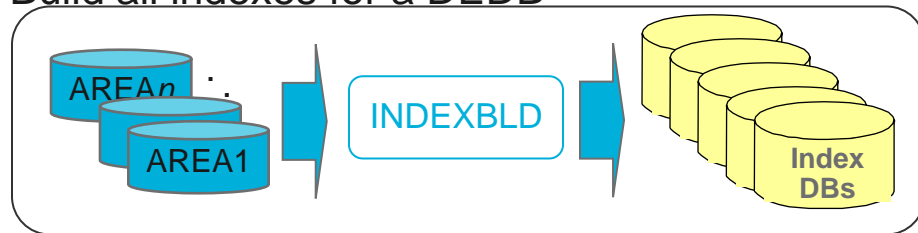
Build selected indexes of a DEDB

Build indexes for specific areas
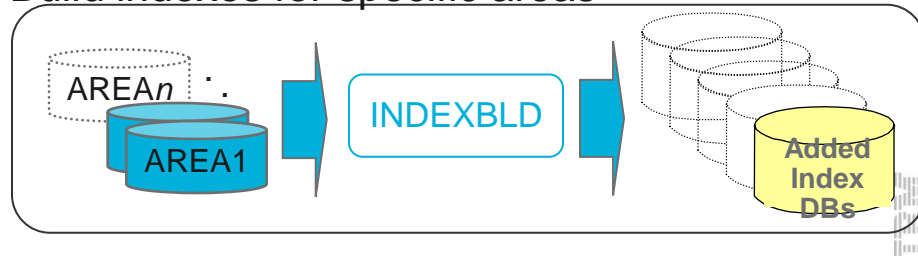
Challenges in FPSI index synchronization

- For fast updates of areas, you may use the FPSI option SETI PSB=xxx
  - This option allows the suppression of index maintenance in a BMP region while a DEDB area is being updated.

- The result is that the secondary indexes no longer accurately reflects changes in the DEDB areas

- So, re-synchronization of the DEDB areas and the secondary indexes is required in the next maintenance window

- But, areas cannot be made offline to build secondary index

Before the BMP application job ran

AREAs   Indexes

Source segment   Pointer segment

BMP Application
Update *without* SETI PSB=xxx

Source segment   Pointer segment
Source segment   Pointer segment

BMP Application
Update *with* SETI PSB=xxx

Source segment   Pointer segment
Source segment

Missing index pointer segment
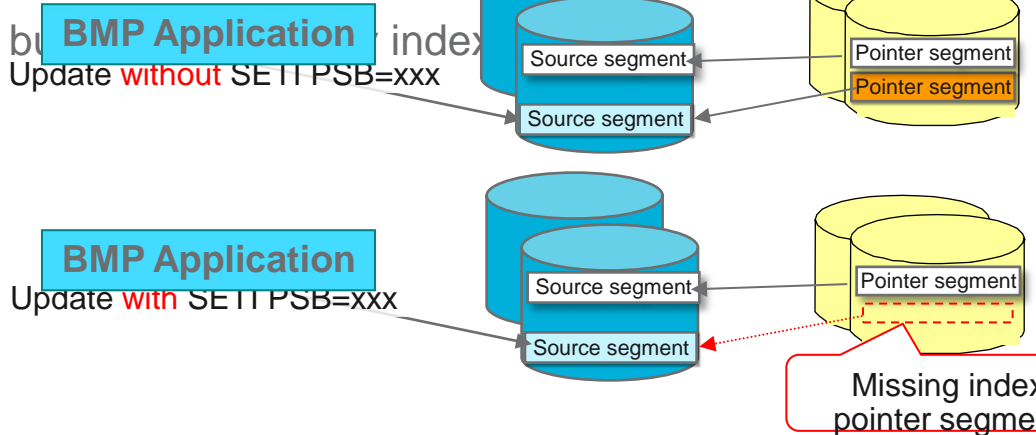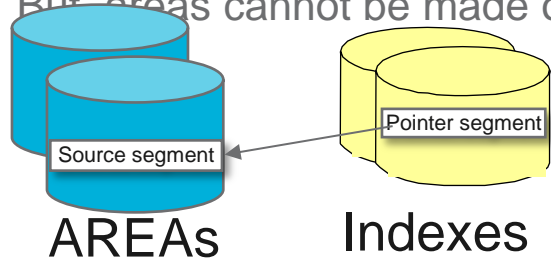
Online Resynchronization solution by FP Solution Pack V1.3

- FP Advanced (FPA) INDEXBLD provides the capability of **Online Resynchronization (Online Resync)** function
  – Online Resync function invokes FPUTIL region internally and issues DL/I calls to update secondary indexes to resynchronize them with the online DEDB areas
  – You do not need to generate a PSB to access secondary indexes

- The Online Resync function is activated by the combination of the following options for the INDEXBLD command:
  – IDXPROC=RESYNC
  – TOICTL=NONE

- The new option **RESYNCMODE** of the INDEXBLD command is used to specify a run mode of the Online Resync function

Online Resync run modes (RESYNCMODE)

- The following four run modes are supported.

DIRECTUPDATE	VERIFYUPDATE	VERIFY	UPDATE
- Scans only the DEDB and then updates the secondary index databases - Scan and update are done in one job	- Scans the DEDB and secondary index databases and then updates the secondary index databases - Scan and update are done in one job	- Scans the DEDB and secondary index databases and then generates Resync Pointer Segment records that are used as input for UPDATE mode	- Reads Resync Pointer Segment records that were generated in VERIFY mode and updates the secondary index databases.

✓ Note: For a secondary index database that contains non-unique keys, RESYNCMODE=DIRECTUPDATE must be used.

Four phases of Online Resync process

Phase 1: **Area Scan processing**
- Scan the online DEDB AREAs and generate pointer segments in parallel

Phase 2: **Verify processing**
- Read pointer segments from the indexes in parallel
- Compare the two sets of pointer segments, then generate Resync Pointer Segment records only for those have differences, in parallel

Phase 3: **Intermediate Data Set Write Processing**
- Write Resync pointer segment records into Resync Pointer Segment record date sets in parallel

Phase 4: **Update processing**
- Update the indexes serially

AREAs

The color and the number in circles indicate processing steps in each run mode

- **DIRECTUPDATE**
- **VERIFYUPDATE**
- **VERIFY**
- **UPDATE**
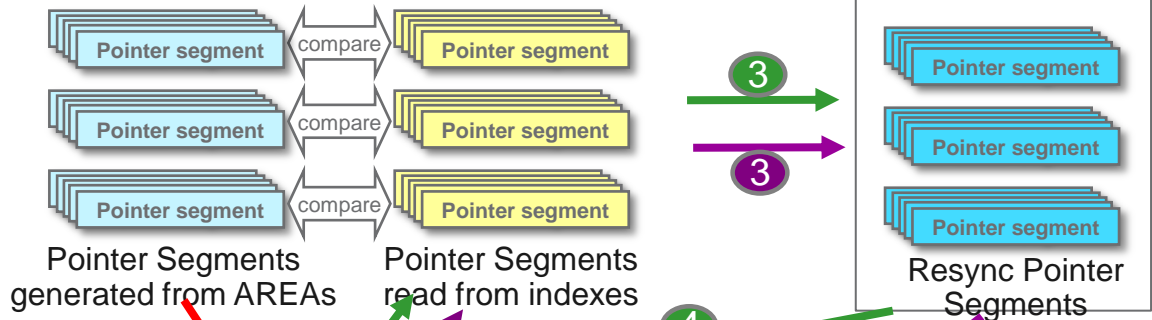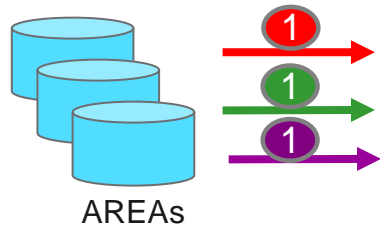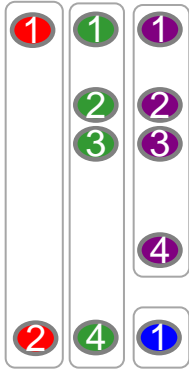
Pointer Segments generated from AREAs

Pointer Segments read from indexes

Resync Pointer Segments

Resync Pointer Segment record data sets

Indexes

Online Resync function JCL examples

DIRECTUPDATE mode

```
//HFPSYSIN DD *
  GLOBAL DBRC=YES,IMSID=IMS1
  INDEXBLD DBD=DEDBJN22,
    IAREA=(AREA1,AREA2,AREA3),
    INDEXDBD=ALL,
    IDXPROC=RESYNC,
    RESYNCMODE=DIRECTUPDATE,
    TOICTL=NONE
/*
```

VERIFY mode

```
//HFPSYSIN DD *
  GLOBAL DBRC=YES
  INDEXBLD DBD=DEDBJN22,
    IAREA=ALL,
    INDEXDBD=(INDEX1,INDEX2,INDEX3),
    IDXPROC=RESYNC,
    RESYNCMODE=VERIFY,
    TOICTL=NONE
  FILECTL
    DSNAME='HFP.&INDEX',
    STORCLAS=STRCLS
/*
```

These statements specify Resync
Pointer Segment record data sets

VERIFYUPDATE mode

```
//HFPSYSIN DD *
  GLOBAL DBRC=YES,IMSID=IMS1
  INDEXBLD DBD=DEDBJN22,
    IAREA=ALL,
    INDEXDBD=ALL,
    IDXPROC=RESYNC,
    RESYNCMODE=VERIFYUPDATE,
    TOICTL=NONE
/*
```

UPDATE mode

```
//HFPSYSIN DD *
  GLOBAL DBRC=YES,IMSID=IMS1
  INDEXBLD DBD=DEDBJN22,
    INDEXDBD=(INDEX1,INDEX2,INDEX3),
    IDXPROC=RESYNC,
    RESYNCMODE=UPDATE,
    TOICTL=NONE,
    RESYNCDSNMASK='HFP.&INDEX'
/*
```

Reference: Resync Pointer Segment record data sets

- Methods for specifying the Resync Pointer Segment record data sets
  1. **Using S0*nnnnn*R DD statements (in VERIFY and UPDATE modes)**
     - *nnnnn* is a sequential number determined by the order the secondary index databases appear on the LCHILD statements in the DBDGEN source for the DEDB
     - An example:
       ```
       DBD  NAME=DEDB0001,ACCESS=DEDB,RMNAME=RMOD1
             ...
       LCHILD NAME=(SEG1,FPSI0001),PTR=SYMB    <- S0000001R for Index FPSI0001
             ...
       LCHILD NAME=(SEG1,FPSI0002),PTR=SYMB    <- S0000002R for Index FPSI0002
             ...
       LCHILD NAME=(SEG1,FPSI0003),PTR=SYMB    <- S0000003R for Index FPSI0003
             ...
       ```
  2. **Using RESYNCDSNMASK keyword (in VERIFY and UPDATE modes)**
     - Specify the data set mask to use when dynamically allocating cataloged pre-existing Resync Pointer Segment record data sets
  3. **Using FILECTL subcommand (in VERIFY mode)**
     - With DISP=NEW, Resync Pointer Segment record data sets are dynamically created, and the records are written in the data sets
     - With DISP=OLD or DISP=SHR, the existing data sets that are specified by the FILECTL subcommand are used

Choice of an Online Resync run mode

- **Case 1:** If you want to update indexes in parallel,
    1. Run Online Resync with RESYNCMODE=VERIFY and IAREA=ALL
        - This job creates Resync Pointer Segment Record data sets
    2. Run multiple Online Resync jobs with RESYNMODE=UPDATE, one job for each index, in parallel
        - In each job, the Resync Ptr Segment Record data sets for the subject index are specified

    Advantages: No contention in index updates; parallel processing of multiple indexes

    Disadvantages: Possible contention in accessing target areas; need for intermediate data sets

    Recommendation: Use this option if you have a lot of index pointer segments

- **Case 2:** If you want to avoid contention in area access at insert of index entry,
    – Run Online Resync jobs with RESYNCMODE=VERIFYUPDATE for disjoint subsets of areas

    Advantages: No contention in the target areas; no need for intermediate data sets

    Disadvantage: Possible contention in updates to an index from multiple jobs

    Recommendation: Use this option if the contention above is not expected or would not affect performance

Example scenarios for updating multiple indexes concurrently

- **Case 1:** Run multiple Resync jobs with VERIFY mode concurrently, then run multiple Online Resync jobs with UPDATE mode concurrently, each UPDATE job re-synchronizes one secondary index data set
  - VERIFY job #1: IAREA=1-5, INDEXDBD=ALL
  - VERIFY job #2: IAREA=6-10, INDEXDBD=ALL
  - UPDATE job #1: IAREA=ALL, INDEXDBD=INDEX1
  - UPDATE job #2: IAREA=ALL, INDEXDBD=INDEX2
  - UPDATE job #3: IAREA=ALL, INDEXDBD=INDEX3
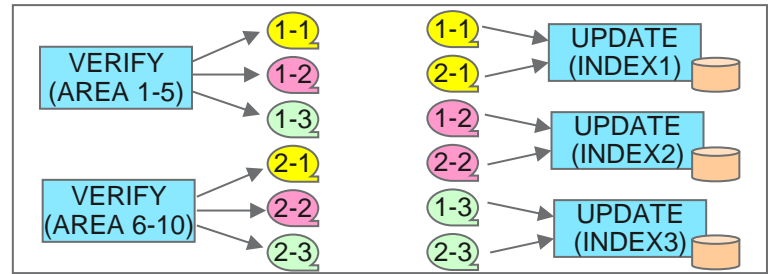
- **Case 2:** Run multiple Resync jobs with VERIFYUPDATE mode concurrently
  - VERIFYUPDATE job #1: IAREA=1-5, INDEXDBD=ALL
  - VERIFYUPDATE job #2: IAREA=6-10, INDEXDBD=ALL
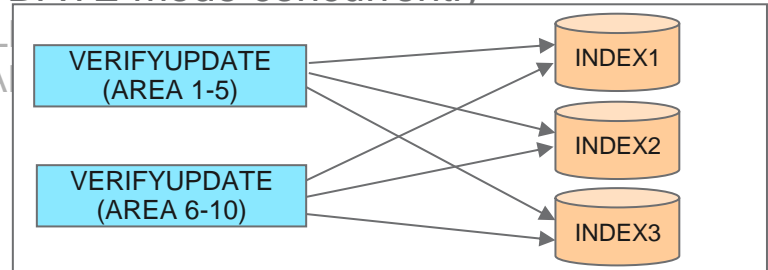
Secondary Index Resync report

▪ Generated by the INDEXBLD command with IDXPROC=RESYNC is specified

▪ Provides the following information:
  – The number of index pointer segments that were inserted to or deleted from each secondary index databases
  – The Resync Pointer Segment record data sets that were created or used

```
IMS HPFP UTILITIES - FPA INDEXBLD                    "Secondary Index Resync Report"                          PAGE:       1
5655-W14 V1R3                                                                                       2014-05-02 10:29:17

   - PRIMARY DEDB NAME            : DEDBJN23
   - NUMBER OF SEC. INDEX DATABASE:     3(PARTITION INCLUDED)
   - RESYNC MODE                  : VERIFYUPDATE
   - TOICTL                       : NONE

SEC. INDEX DATABASE INFORMATION:

  *: RESYNC PERFORMED #: ISRT/DLET FAILED

                                                                          POINTER SEGMENT
 PARTITION                                                 -------------------------------------------------------------
 GROUP     DBDNAME  DDNAME   DSNAME                         BEFORE RESYNC AFTER RESYNC  INSERTED       DELETED
 --------- -------- -------- --------------                 ------------- ------------ ------------- -------------
           *DEDBGS22 GS22KSDS HPFP.GS22KSDS
                                                  (TOTAL)     1,500,000    2,000,000     500,000               0
 --------- -------- -------- --------------                 ------------- ------------ ------------- -------------
 DEDBGS25 *DEDBGS25 GS25KSDS HPFP.GS25KSDS
                                                  (TOTAL)     1,000,000      900,000     200,000         300,000
 --------- -------- -------- --------------                 ------------- ------------ ------------- -------------
 DEDBGS25  DEDBGS26 GS26KSDS HPFP.GS26KSDS
                                                  (TOTAL)       400,000      400,000           0               0
 --------- -------- -------- --------------                 ------------- ------------ ------------- -------------
 DEDBGS25                              (GROUP TOTAL)    1,400,000    1,300,000     200,000         300,000

RESYNC POINTER SEGMENT RECORD DATA SET INFORMATION:

   INDEX
```

Resync Pointer Segment Dump report

- Generated by the INDEXBLD command is specified with IDXPROC=RESYNC and **RESYNCDUMP=YES**
  - The maximum number of pointer segments printed in the report is controlled by the **RESYNCDUMPMAX** keyword

- Provides the following information:
  - The index pointer segments that are inserted to or deleted from the secondary index databases

```
IMS HPFP UTILITIES - FPA INDEXBLD                        "Resync Pointer Segment Dump report"                                    PAGE:      1
5655-W14 V1R3                                                                                                               2014-05-02 10:29:17

DBDNAME: DEDBGS22   DDNAME: GS22KSDS   DSNAME: HPFP.GS22KSDS
 DELETED  (THE RELATED AREA IS AREA NO:     1   AREANAME: DB23AR0 )
   ------------ ----  ----------------------------------------------------------------  --------------------------------
   SEARCH        0000  C1F1F2                                                            *A12                           *
   SUBSEQ        0003  F0F0F0F0 F1F2F1F1                                                 *00001211                      *
   DDATA         000B  D1F1F2                                                            *J12                           *
   SYMBOLIC PTR  000E  F1F0F0F0 F0F0F0F0 F2F1F0F0 F0F0F0F1 F2F1                          *100000002100000121            *
   ------------ ----  ----------------------------------------------------------------  --------------------------------
 INSERTED (THE RELATED AREA IS AREA NO:     1   AREANAME: DB23AR0 )
   ------------ ----  ----------------------------------------------------------------  --------------------------------
   SEARCH        0000  C1F1F3                                                            *A13                           *
   SUBSEQ        0003  F0F0F0F0 F1F2F1F1                                                 *00001211                      *
   DDATA         000B  D1F1F3                                                            *J13                           *
   SYMBOLIC PTR  000E  F1F0F0F0 F0F0F0F0 F2F1F0F0 F0F0F0F1 F2F1                          *100000002100000121            *
   ------------ ----  ----------------------------------------------------------------  --------------------------------
 NOT INSERTED BECAUSE PRIMARY DEDB WAS UPDATED  (THE RELATED AREA IS AREA NO:     2   AREANAME: DB23AR1 )
   ------------ ----  ----------------------------------------------------------------  --------------------------------
   SEARCH        0000  C1F2F1                                                            *A21                           *
   SUBSEQ        0003  F0F0F0F0 F2F1F1F1                                                 *00002111                      *
   DDATA         000B  D1F2F1                                                            *J21                           *
   SYMBOLIC PTR  000E  F2F0F0F0 F0F0F0F0 F1F1F1F0 F0F0F0F2 F1F1                          *200000001100000211            *
```