IMS 12 TM Enhancements & IMS 12 Enhancements and the Repository

Suzie Wendler, Diane Goff, Angie Greenhaw
IBM

Wednesday, August 10, 2011
Share Session 9400

Agenda

- IMS TM Enhancements – Suzie Wendler (wendler@us.ibm.com)
- IMS System Enhancements – Diane Goff (dgoff@us.ibm.com)
- The Repository – Angie Greenhaw (greenhaw@us.ibm.com)

IMS TM Enhancements

- APPC and OTMA Shared Queues Enhancement
  - Removes the dependency on RRS for Synclevels None|Confirm
- OTMA Enhancements
- WMQ Message Expiry Support
- IMS Connect Enhancements
- IMS-IMS Connectivity
  - OTMA Support for Asynchronous IMS-IMS Communications
  - MSC TCP/IP Support

APPC and OTMA SQ Enhancement

- New capability that removes the dependency on RRS in a Shared Queues environment for
  - APPC synchronous conversations and OTMA CM1 (send-then-commit) Applies only to synclevel=None | Confirm
    - Synclevel=Syncpoint still requires RRS
  - Communications use XCF services
  - New options for the existing AOS= parameter in DFSDCxxx
- Benefit
  - Using XCF rather than RRS allows IMS to be the syncpoint manager
    - Enhances the performance of the commit processing by eliminating
      - RRS logging overhead
      - Potential RRS commit processing bottleneck
      - Overhead associated with communicating with an external syncpoint manager

OTMA ACEE Reduction for Multiple OTMA Clients

- New capability that creates, shares and **caches** a **single** ACEE associated with a RACF userid
  - Shared across multiple OTMA member clients (TMEMBER)

- AND… a new maximum ACEE aging value during client-bid
  - 999999 seconds (11.5 days)
    - Previously 68 years
    - Range: 300 seconds to 999999 seconds
      - If OTMA receives a value less than 300, the value is reset to 0 and OTMA will not refresh ACEEs

- A cached ACEE has an aging value based on the OTMA member client with the lowest value

Benefits of OTMA ACEE Enhancements

- Cached ACEEs
  - Reduce the system storage requirements while providing better security and performance
    - Only one copy of the ACEE instead of multiple per OTMA client
      - Reduced storage usage
      - Reduced security exposure
      - Improved performance
  - Provide consistency
    - Same security result regardless of which OTMA client is used
- Lower maximum ACEE aging value
  - Triggers faster ACEE cache refresh
    - Reduces security exposure, e.g., userid is revoked or access permissions are changed

OTMA Performance

- Reduced path length for OTMA transaction processing
  - Simplification in logic when validating a TPIPE name
    - Only when a new tpipe name is received on a message
      - Instead of when each message is received
  - APARs PM20292 (V10) / PM20293 (V11)
    - Shipped with the ICAL enhancements
- Benefit
  - Improved OTMA performance

V11 Transacton Expiration SPE

- IMS Transaction Expiration SPE

- APARs PM05984 (IMS10) / PM05985 (V11)
  - Sends DFS3688I message instead of DFS555I or DFS2224I message for transaction expiration during application GU phase

    DFS3688I Transaction *aaaaaaaa* expired: EXPRTIME=*nnnnnn*, ELAPSE=*ssssss*

    Tmember xxxxx Tpipe xxxx

  - Enhancement only affects **OTMA** messages
    - Expired non-OTMA messages already receive DFS3688I
      - PK86426/UK47070 (V11) – non-OTMA transaction expiration is V11 only

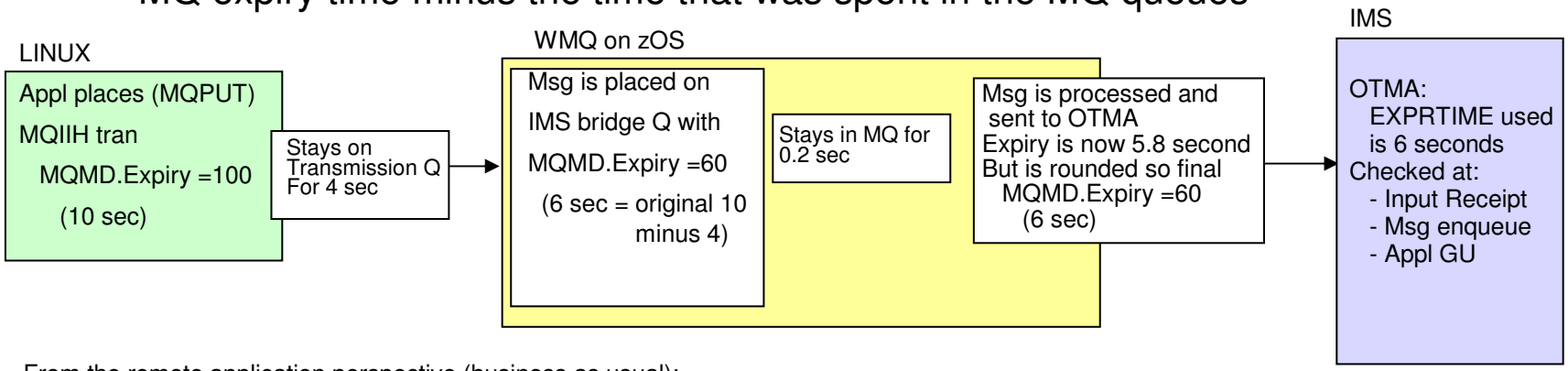
        DFS3688I Transaction *aaaaaaaa* expired: EXPRTIME=*nnnnnn*, ELAPSE=*ssssss*

MQ Message Expiration

- Extension of the WebSphere MQ (WMQ) Message Expiry facility to include the IMS transaction expiration function (WMQ 7.01)
  - A new service parameter
    - CSQ6SYSP SERVICE = 0000000001 or also specified through the SET SYSTEM SERVICE(0000000001) command
      - Used in conjunction with other queue manager service parameters
        - e.g. if queue manager already uses service parm 0040 then setting the new service would result in 0040000001
      - Provides toleration of an OTMA NACK_FOR_TRANS_EXPIRED response from IMS through the OTMA support
        - Leverages WMQ expiry processing as if the message had expired prior to sending the message to OTMA

MQ Message Expiration …

- User-Specified Expiry time (message-level)
  - A value is passed to IMS if an MQ message expiry time (MQMD.Expiry) exists for the message AND the service parameter is set
    - Value is in 10ths of a second
    - **The residual expiry time for the message is built into the OTMA interface**
      - MQ expiry time minus the time that was spent in the MQ queues

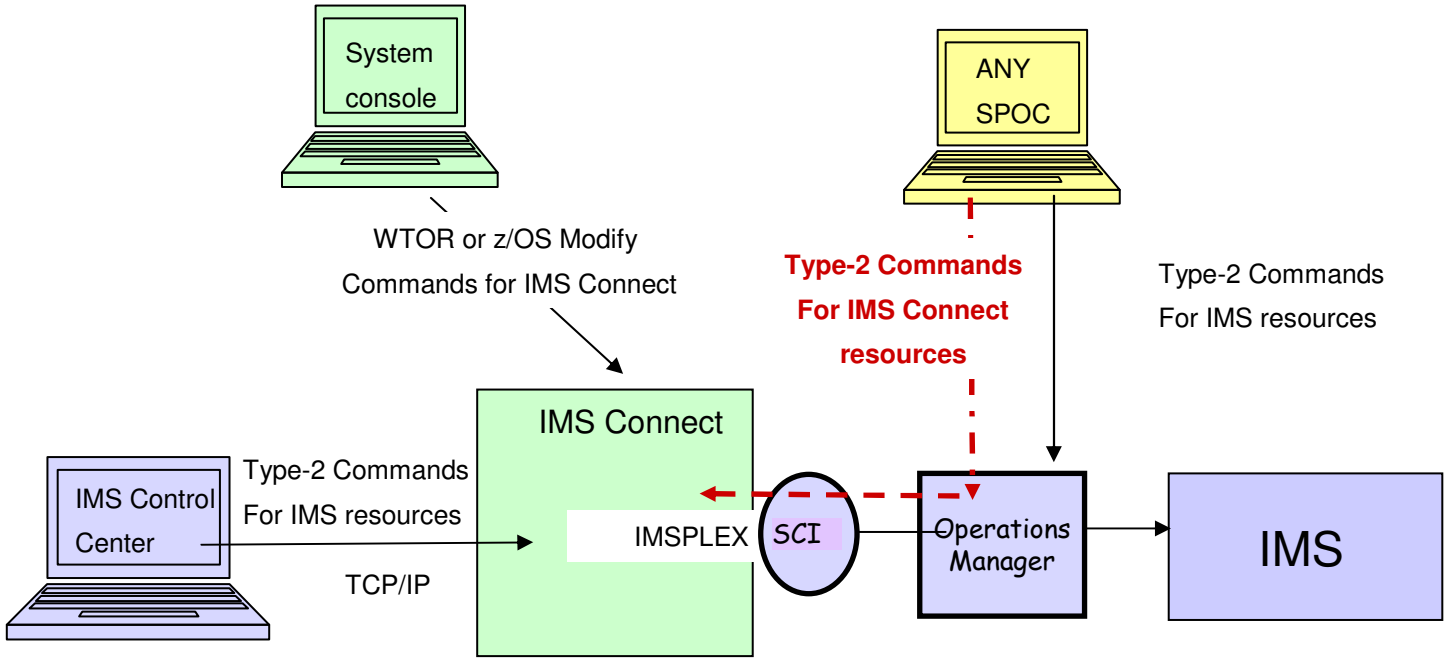


From the remote application perspective (business as usual):

- The MQPUT application will be unaware of an expiry unless it specifies a Report option which can
  > include the generation of an expiry report which will be sent to the specified reply-to queue,
  > passing the remaining expiry interval from a request message to a response message,
  > or just discarding the expired message.

New Type-2 Commands for IMS Connect

- New Type-2 commands for IMS Connect resources
  - QUERY IMSCON
  - UPDATE IMSCON



XML Converter Refresh

- New Command to refresh an XML converter file that is already in use

```
UPDATE IMSCON TYPE(CONVERTER)…
xx,REFRESH CONVERTER NAME(cvtrname)
F hws,UPDATE CONVERTER NAME(cvtrname) OPTION(REFRESH)
```

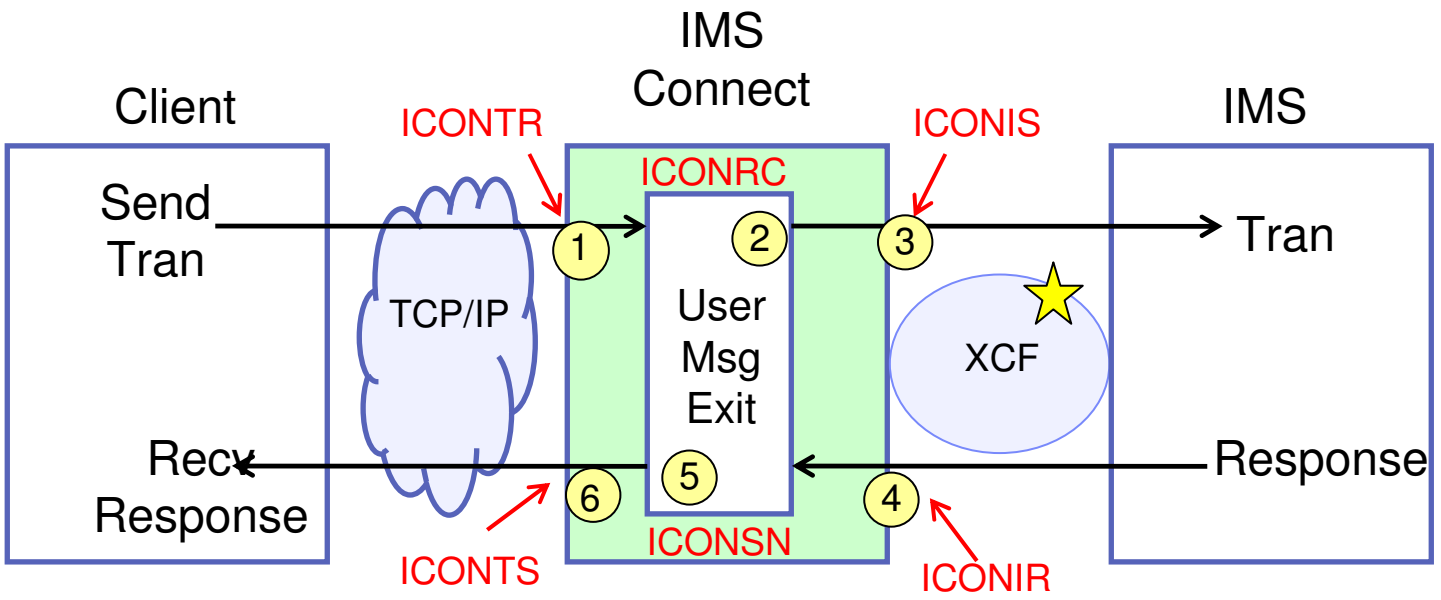
  - Supported by all command interfaces: Type-2, WTOR, z/OS Modify

  - Converter files continue to be:
    - Generated using RDz
    - Loaded by IMS Connect from STEPLIB/JOBLIB/LNKLST

- Benefit
  - More timely ability to change and implement converter files
    - Without requiring an IMS Connect restart

New IMS Connect Recorder Trace Points

F HWS1,UPDATE TRACETABLE NAME(RCTR) OWNER(HWS) **LEVEL(HIGH)** EXTERNAL(YES)



No ICONIS/ICONIR support for the SCI interface (type-2 commands and ODBM)

1. ICONTR – Receive from TCP/IP
2. ICONRC – User Msg Exit Receive
3. ICONIS – Send to IMS
6. ICONTS – TCP/IP Send to Client
5. ICONSN – User Msg Exit XMIT
4. ICONIR – Receive from IMS

- Benefit
  - Additional trace points provide the ability to capture client errors for improved problem determination and analysis
  - The use of BPE external tracing allows large amounts of data to be captured

IMS Connect – RACF Userid Caching

- Existing IMS Connect security with RACF=Y
  - Limited caching of RACF Utoken
    - Consecutive requests on a persistent socket with the same Userid/Password/Group
- IMS 12 enhancement with RACF=Y
  - Common cache for userids across ALL sessions and ALL ports
    - HWSCFG HWS statement: UIDCACHE={N|Y} , UIDAGE=aging_value

```
xx,VIEWHWS
                    HWSC0001I   HWS ID=HWS1      RACF=Y  PSWDMC=R
                    HWSC0001I      UIDCACHE=Y   UIDAGE=300
                    HWSC0001I      MAXSOC=2000  TIMEOUT=6000
                    HWSC0001I      NUMSOC=6      WARNSOC=80%  WARNINC=5%
                    HWSC0001I      RRS=Y   STATUS=ACTIVE
                    HWSC0001I      VERSION=V12 IP-ADDRESS=009.030.218.050
                    HWSC0001I      SUPER MEMBER NAME=      CM0 ACK TOQ=
                    HWSC0001I    ADAPTER=Y
```

CM0 ACK NoWait for RYO Clients

- Existing protocol for Roll Your Own (RYO) clients requires
  - CM0 Send-Receive interactions to receive a timeout notification after ACK/NAK
    - Receive and timeout flow adds unnecessary overhead to the client application
- New option of NoWait on ACK or NAK
  - Indicates the remote client will not issue subsequent receive

Previous CM0 send-receive flow

Send request
Receive response
Send ACK
Receive T/O

New CM0 send-receive flow

Send request
Receive response
Send ACK NoWait

(no need to issue receive for final timeout)

- Benefit
  - Greater efficiency and simplified interaction
    - Eliminates need for extra send after an ACK/NAK

Partial Read Status

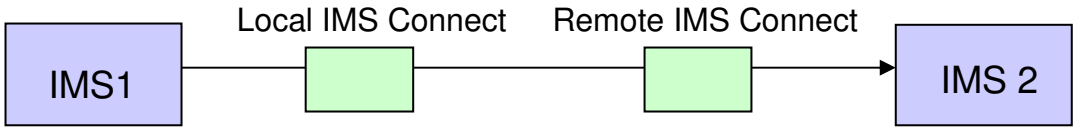
- New READ client status
  - The message has been received by IMS Connect but is not yet considered a complete input message
    - Should be transient but can be an indicator of a problem
    - Affects VIEWPORT, VIEWHWS, QUERY MEMBER, QUERY PORT, QUERY IMSCON command output
- Benefit
  - Facilitates the detection of a remote application programming error
    - Invalid length specification of an input message

IMS Connect User Exit Load Modules

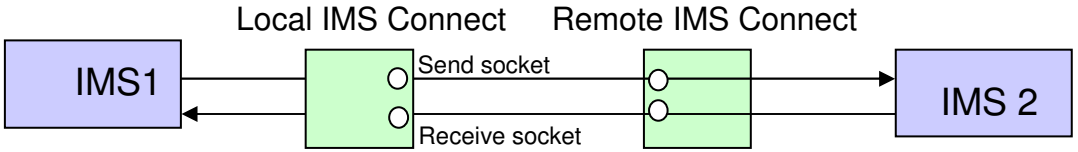
- IMS Connect ships load modules for User Exits
  - HWSUNIT0, HWSJAVA0, HWSSMPL0, HWSSMPL1
    - Previously, working samples were provided but always had to be assembled and bound
      - Even if no changes were made to the provided source samples
- Benefit
  - Eases installation and maintenance processing if the user exits are to be used unchanged

IMS to IMS TCP/IP Connectivity

- Enhancements to leverage TCP/IP networks for communications between IMS systems for:
  - OTMA Support for Asynchronous IMS-IMS Communications
    - Uses one-way message communications (ALTPCB)

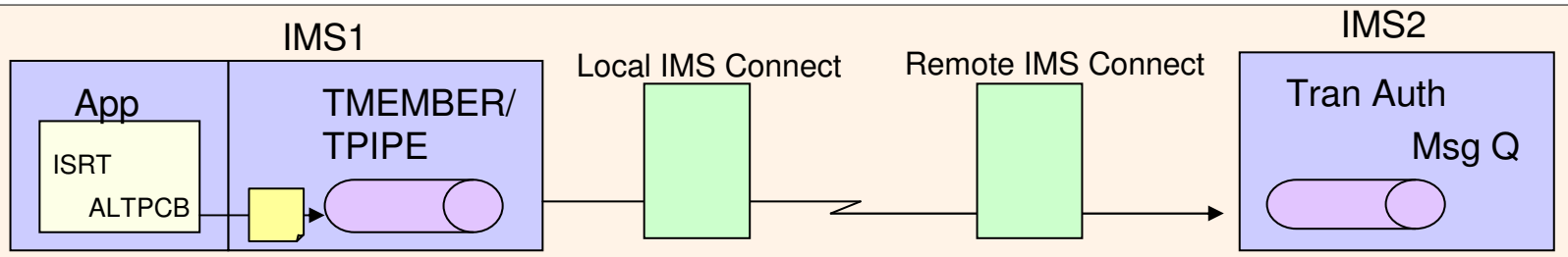


  - TCP/IP-Type Physical Links (MSC)
    - Request and response message communications
      - IMS Connect processes both the request and response messages as one-way asynchronous messages



Asynchronous IMS-IMS TCP/IP Support …

- OTMA
  - Sends OTMA remote ALTPCB messages to IMS Connect using new destination information
    - OTMA destination descriptors or DFSYDRU0 exit Routine
- IMS Connect
  - Receives OTMA ALTPCB messages from a local IMS and sends them to the remote IMS Connect for processing in the remote IMS
    - Enhanced IMS Connect configuration specifications



ALTPCB destination is resolved using an OTMA Destination Descriptor or the DFSYDRU0 exit

IMS Connect configurations defines the connection to the remote Partner

Usage and Benefits

- Usage
  - IMS applications: ISRT ALTPCB
  - IMS environment: destination descriptor or a DFSYDRU0 exit routine
  - IMS Connect: configuration specifications
- Benefits
  - Supports TCPIP communications to invoke transactions between IMS systems without having to create or maintain a separate gateway solution
    - IMS-provided and supported solution

MSC TCP/IP

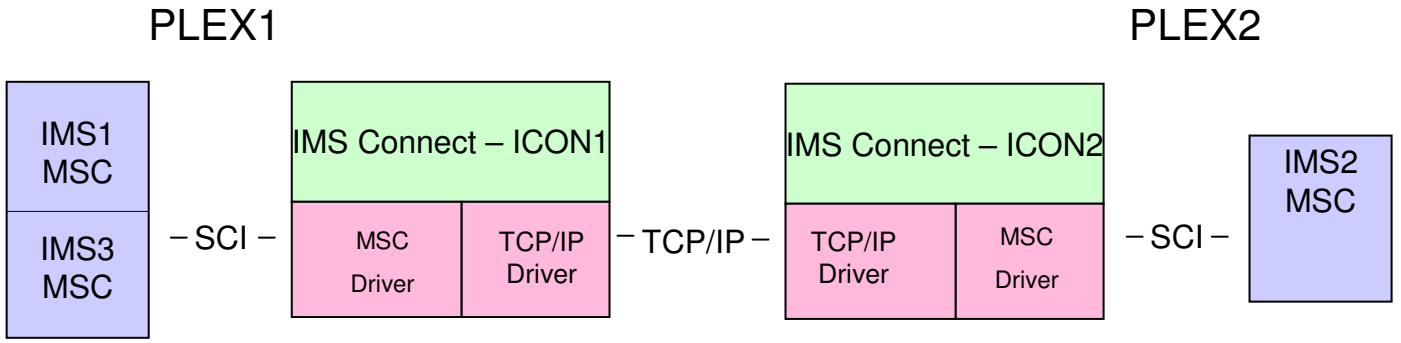
- Support for MSC communications across a TCP/IP network
  - MSC TCP/IP leverages IMS Connect and the Common Service Layer
  - A new physical link MSPLINK TYPE=TCPIP
    - Provides a mechanism to
      - Take advantage of TCP/IP networks
      - Complement or backup existing SNA/VTAM links
      - Take advantage of potentially higher bandwidths
    - Supports operational compatibility with other link types (CTC, MTM, VTAM)
      - Starting, stopping, updating, displaying, and assigning resources
  - Only between IMS 12 systems

MSC TCP/IP

- MSC TCP/IP leverages IMS Connect and the Common Service Layer
  - IMS Connect sends/receives messages via the TCP/IP network
    - IMS Connect manages the TCP/IP communications
    - IMS MSC manages the message processing
  - CSL provides the Structured Call Interface (SCI) for communications between IMS components including IMS Connect
    - Each IMS and its local MSC-routing IMS Connect system must be part of the same IMSplex
      - IMSPLEX= plexname parameter in the Common Layer Section of the DFSDFxxx of IMS proclib
    - The Operations Manager (OM) is not required but recommended
      - For type-2 command support

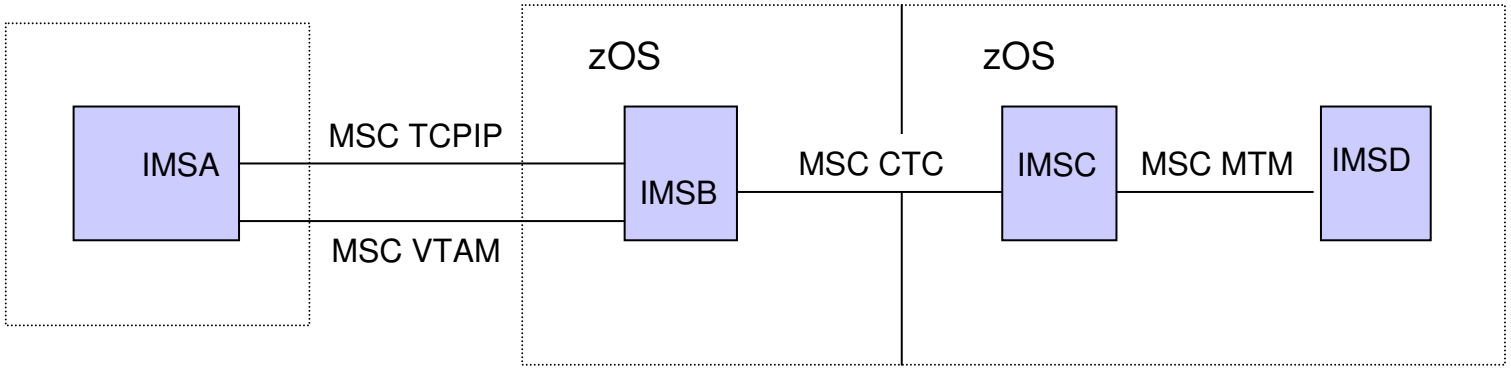
MSC TCP/IP …

- IMS to IMS Connect functionality
  - Isolates TCP/IP from the IMS Control Region
    - Uses the existing IMS Connect TCP/IP support
  - Provides a new MSC driver as well as TCP/IP driver for MSC
  - Supports communication with IMS via the Structured Call Interface (SCI)



MSC TCP/IP - Benefits

- Benefits
  - Takes advantage of TCP/IP networks for MSC
    - Can potentially provide for a higher MSC bandwidth
  - Supports different configurations
    - Coexistence with or backup of VTAM/SNA links
    - Increases availability
      - Logical links can be moved between VTAM and TCPIP
    - Flexibility



TM Summary

- IMS 12 continues to enhance:
  - APPC
  - OTMA
  - IMS Connect

- And introduces new ways to support
  - IMS-IMS Communications

IMS 12 System Enhancements

IMS 12 System Enhancements

- Dynamic resource definition (DRD) enhancements
- Extended address volume (EAV) enhancement
- IMS logger enhancements
- System pools storage enhancement
- Command enhancements
- Syntax checker enhancements
- CQS traceability enhancements

Dynamic Resource Definition (DRD) Enhancements

- New UPDATE option for IMPORT command
  - Previously, IMPORT could only be used for adding runtime resource definitions/descriptors that did not exist in the target IMS system
  - New IMPORT .. OPTION(UPDATE) allows existing runtime resource definitions/descriptors in the the target IMS to be changed
    - Command fails if changed definition is in use
- DRD usage of the IMS repository function
  - Previously, stored resource definitions/descriptors were kept in resource definition data sets (RDDSs)
  - New IMS repository function provides an additional method for storing stored resource definitions/descriptors
  - IMS repository will be described later in this session
- Benefits
  - Improved manageability for DRD

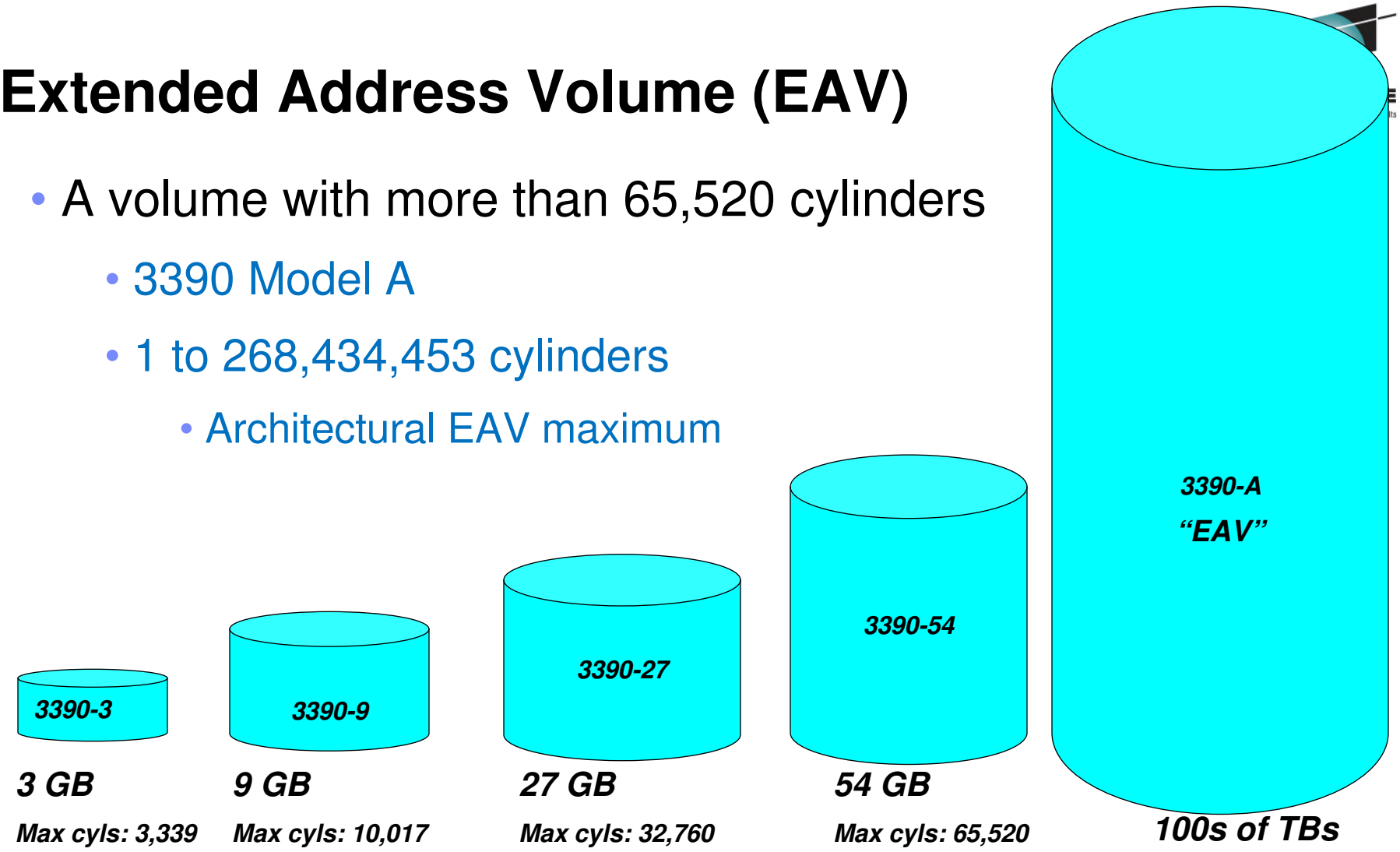
Extended Address Volume (EAV) Enhancement

- IMS 12 allows certain non-VSAM IMS data sets to use EAV volumes
  - Data sets can reside in Extended Address Space (EAS) on EAV volumes
  - z/OS addressable disk storage increased beyond 65K cylinders
    - New architecture will support 100's of Terabytes on single volume
    - Storage is addressed using new 28-bit cylinder/track address
  - Requires z/OS 1.12

- IMS 11 provided support for IMS VSAM data sets to use EAV volumes

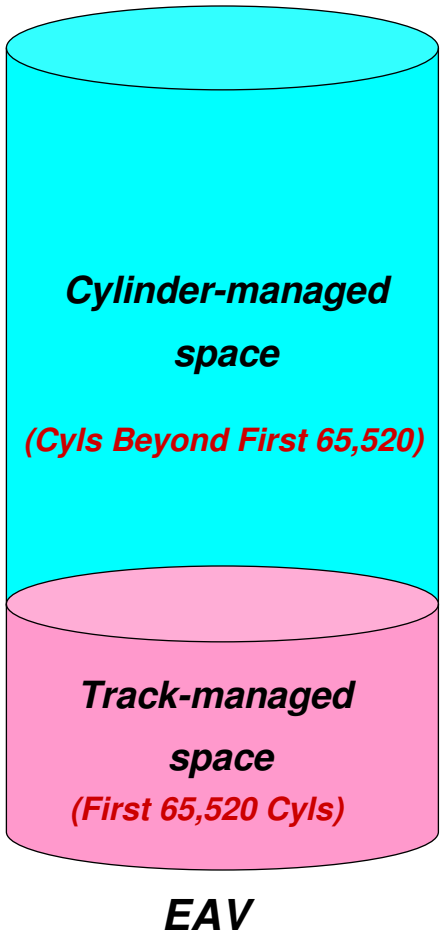
Extended Address Volume (EAV)

- A volume with more than 65,520 cylinders
  - 3390 Model A
  - 1 to 268,434,453 cylinders
    - Architectural EAV maximum



Maximum Sizes

EAV Key Design Points



- EAV maintains 3390 track format
  - Track-managed space:
    - Area on EAV within the first 65,520 cyls
    - Space allocated in track or cyl increments
    - Storage for “small” data sets
  - Cylinder-managed space:
    - Area on EAV located above first 65,520 cyls
    - Space is allocated in multicylinder units
    - Storage for “large” data sets
  - New DSCB format types identify EAS data sets
    - New formats (Format 8 and 9) in VTOC
    - Data set resides in cylinder-managed space

Non-VSAM IMS Data Sets Supported

- Overflow Sequential Access Method (OSAM) data sets
  - OSAM database data sets
  - Restart data set (RDS)
  - Message queue blocks data set
  - Long and short message data set
- IMS Online Log Data Sets (OLDS)
- IMS Write Ahead Data Sets (WADS)
- IMS SPOOL data sets
- BPE External Trace Data Sets

Extended Address Volume (EAV) Enhancement for non-VSAM data sets

- Prerequisites
  - Software requirements
    - z/OS 1.12
  - Hardware requirements
    - DS8000, DS8700
    - 3390 Model A

- Benefits
  - Supports the placement of more data sets on a single volume
  - Allow users to manage fewer numbers of larger volumes
  - Less need for multi-volume OSAM

IMS Logger Enhancements …

- Extended Format Support for OLDS and SLDS (optional)
- Option for log buffers above the 2-gigabyte boundary (“bar”) in virtual
- WADS management changed to be more efficient

IMS Logger Enhancements …

- New optional capability for OLDS and SLDS
  - IMS 12 allows OLDS and SLDS to be defined as extended format data sets
  - Use of extended format data sets allows striping
    - Striping allows multiple concurrent I/Os for sequential processing
      - Data set is spread across multiple volumes
      - Increased logging rates
- Option for log buffers above the 2-gigabyte boundary (“bar”) in virtual
  - Frees substantial amount of ECSA
  - OLDS must be in extended format with BLKSIZE 4K multiple
  - BUFSTOR=64 on OLDSDEF statement in DFSVSMxx

IMS Logger Enhancements

- WADS management changed to be more efficient
  - Track groups no longer used
  - WADS written in wrap around fashion
  - WADS should be sized to provide enough space for any OLDS buffers not yet written at any time plus one track
  - WADS should be kept in cache in storage subsystem

- Benefits
  - Increased logging bandwidth / improved logging performance
  - ECSA constraint relief
  - Simplified WADS management for improved performance

System Pools Storage Enhancement

- Storage for selected database pools is obtained in 31-bit virtual storage, backed by 64-bit real storage
  - DBWP – Database work pool
  - DLDP – DMB pool
  - DLMP – CSA PSB pool
  - DPSB – DLI PSB pool
  - PSBW – PSB work pool
- Benefits
  - Reduction in 31-bit fixed real frames for fixed pools
  - Some users will now be able to fix these pools
    - Previously, they were constrained by 31-bit real storage

Command Enhancements

- Enhancements to existing commands and new commands
  - CQS trace command enhancements
  - DBRC command enhancements
  - Dynamic database buffer pool command enhancements
  - Dynamic resource definition (DRD) command enhancements
  - Fast Path secondary index command enhancements
  - HALDB command enhancements
  - IMS Connect command enhancements
  - MSC command enhancements
  - IMS repository function command enhancements
  - OTMA command enhancements

Command Enhancements

- Enhancements are focused on type-2 commands for the Operations Manager (OM) environment

- Benefits
  - Support of new IMS 12 functions
  - Improved manageability

Syntax Checker Enhancements

- Syntax Checker supports PROCLIB members for IMS 12 / IMS 11 / IMS 10
  - IMS 9 PROCLIB members are not supported
- All previously supported members are supported
  - Newly added parameters of these members are supported
- Support added for Repository Server configuration member
- Support added to view/save parameters of members in a custom order
  - Formerly, only alphabetical order was used

- Benefits
  - Support of new IMS 12 PROCLIB members
  - Improved usability with custom order of parameters

CQS Traceability Enhancements

- Existing CQS structure trace table (STR) can quickly fill, wrap around, and lose critical trace entries
- Two new BPE trace tables are available to track CQS structure events
  - One for overflow events (OFLW), one for structure events (SEVT)
  - Retain critical trace data for longer periods of time

- Benefits
  - Improves CQS serviceability

IMS 12 System Enhancements

- Dynamic resource definition (DRD) enhancements
- Extended address volume (EAV) enhancement
- IMS logger enhancements
- System pools storage enhancement
- Command enhancements
- Syntax checker enhancements
- CQS traceability enhancements

The Repository

IMS Repository Agenda

- Overview of the IMS repository function
- IMS repository function components
  - Repository Server (RS) address space
  - Repository data sets
  - Repository catalog data sets
  - CSL requirements and RM usage
- IMS repository setup
  - Repository Server setup
  - Creating/enabling an IMSRSC repository for DRD
- IMS repository commands and usage
- Migration to DRD with the repository

IMS Repository Function Overview

- A ‘repository’ is a generalized data storage facility that can be used to store various types of information
- The IMS repository function is a centralized method for storing and retrieving resource definitions in an IMSplex
  - Enables multiple IMS systems in a multiple-IMS IMSplex to manage, store, share, and retrieve resource definitions
  - Enables a single IMS system in a single-IMS IMSplex to manage, store, share, and retrieve resource definitions
- Focus is on improving the systems management and resource management aspects of handling IMS resource definitions
  - Across multiple IMSs or for a single standalone IMS
  - For test systems, for production systems

IMS Repository Function Usage

- In IMS 12, the resource and descriptor definitions for Dynamic Resource Definition (DRD) can be stored in an IMS repository
  - Contains resource definitions for programs/transactions/databases/FP routing codes & descriptors
  - Called the IMSRSC (IMS resource) definition repository
  - Provides an alternative to using RDDSs (resource definition data sets) for DRD
    - Replaces one or more sets of RDDSs in an IMSplex with a single repository
  - Eliminates the need to manually coordinate and manage separate RDDSs per IMS across a multiple-IMS IMSplex
  - Provides an alternative to using MODBLKS with SYSGEN and online change
  - Considered a strategic alternative to the RDDS
- IMS 12 can retrieve the stored resource definitions from the IMSRSC repository to dynamically generate runtime resources for DRD

IMS 12 Support for the DRD Function …

- DRD users in IMS 10 and IMS 11 moving to IMS 12
  - Can use existing RDDSs from IMS 10 or IMS 11 for stored resource definitions in IMS 12
  - Can use existing RDDSs from IMS 10 and IMS 11 for stored resource definitions at initial migration to IMS 12, then can migrate to the new IMSRSC repository
  - Can use the new IMSRSC repository to store definitions in IMS 12

IMS 12 Support for the DRD Function

- Users in IMS 10 and IMS 11 without DRD
  - Can use the new IMSRSC repository for stored resource definitions in IMS 12
  - Can create new RDDSs for stored resource definitions in IMS 12
- Both RDDSs (system and non-system) and the IMSRSC repository can exist together during migration to the IMSRSC repository

IMS Repository Function Benefits

- Consolidation of resource definitions in a single place, the repository
- DRD definitions are the initial implementation of the IMS repository function (to replace RDDSs)
- Full support for populating, managing, storing, sharing, and retrieving a consistent set of DRD stored resource definitions for multiple-IMS IMSplexes and single-IMS IMSplexes
- Manual coordination of multiple RDDSs in a multiple-IMS IMSplex eliminated, replaced by basic functioning of the IMS repository
- Improvements in IMSplex systems and resource management with the repository
- A strategic direction for IMS architecture

IMS Repository Function Components …

- Repository Server (RS)
- Repositories
  - Catalog repository
  - IMSRSC repository(s)
- Common Service Layer (CSL) IMSplex configuration consisting of
  - Operations Manager (OM)
  - Resource Manager (RM)
  - Structured Call Interface (SCI)
  - SPOC for entering type-2 commands
  - Optional resource structure with CQS address space
- Batch utilities
  - Batch ADMIN utility
  - RDDS to / from repository utilities

IMS Repository Function Components …

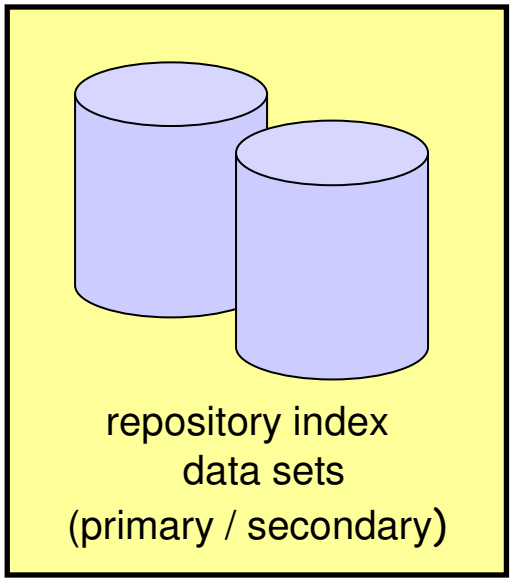
- Repository Server (RS)
  - New BPE-based address space
    - Managed by the RM CSL address space
  - Two types
    - Master Repository Server
      - Single instance
      - Manages access to repository data sets
      - First RS address space to access repository
    - Subordinate Repository Server
      - One or more instances
      - Used if master Repository Server goes down
      - Optional but recommended

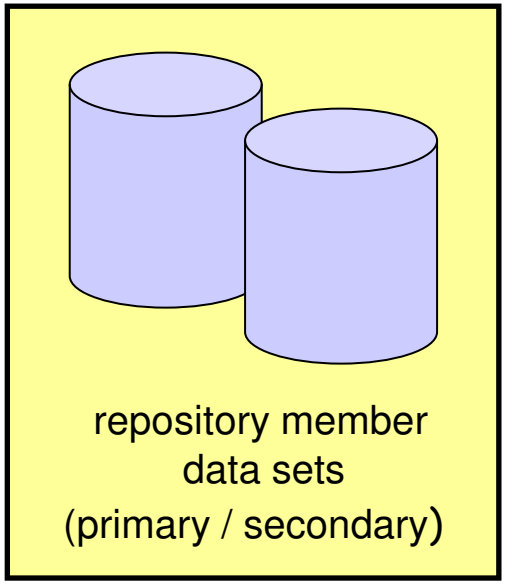
IMS Repository Function Components …

- Repository Server (RS)
  - Uses VSAM KSDS data sets to store information
  - Recommendation
    - One master Repository Server address space per IMSplex
  - Has its own internal repository called the ‘catalog repository’
  - Manages IMS repositories (IMSRSC for DRD)
  - Ensures repository data integrity
  - Uses SAF to restrict access to repositories
  - Provides an audit trail using the z/OS logger
  - Provides tracing capabilities via BPE

IMS Repository Function Components …

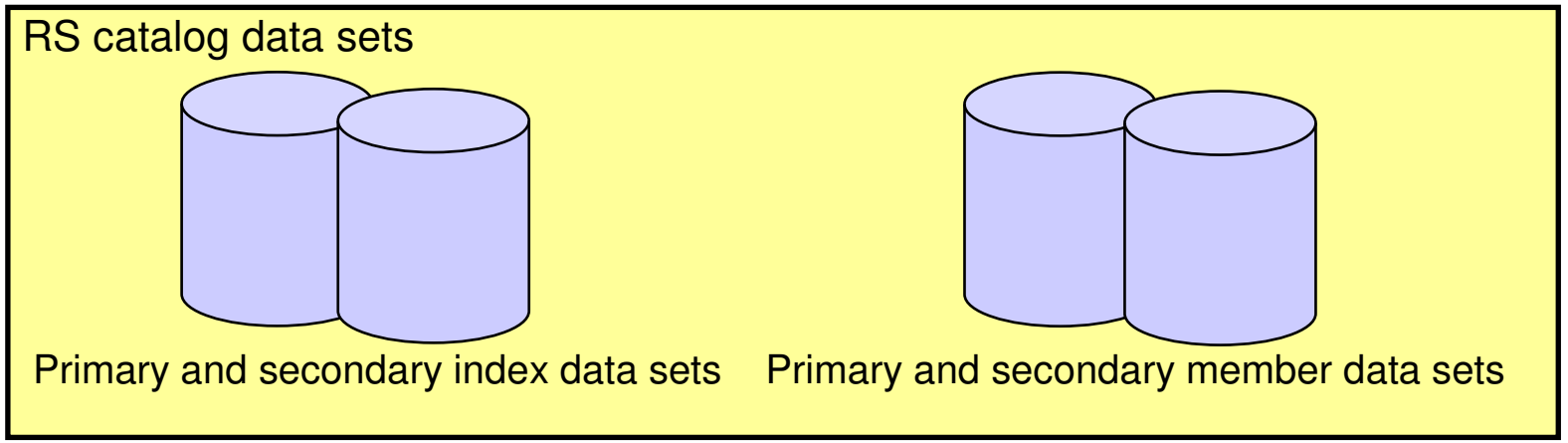
- Repository data sets
  - Multiple sets of VSAM KSDS data sets
  - Each set composed of
    - Repository index data set
    - Repository member data set
    - Each of these has a primary and secondary data set (duplexed)
      - Optional spare set (third) can be defined
  - Two types of repository data sets
    - IMS repository data sets
    - Catalog repository data sets





IMS Repository Function Components …

- Catalog repository (RS catalog data sets)
  - Required per Repository Server
  - Manages the Repository Server (RS) functions
  - Manages information about IMS repository data sets
  - Composed of two pairs of data sets
    - Primary index data set and primary member data set (required)
    - Secondary index data set and secondary member data set (required)
    - No spare capability

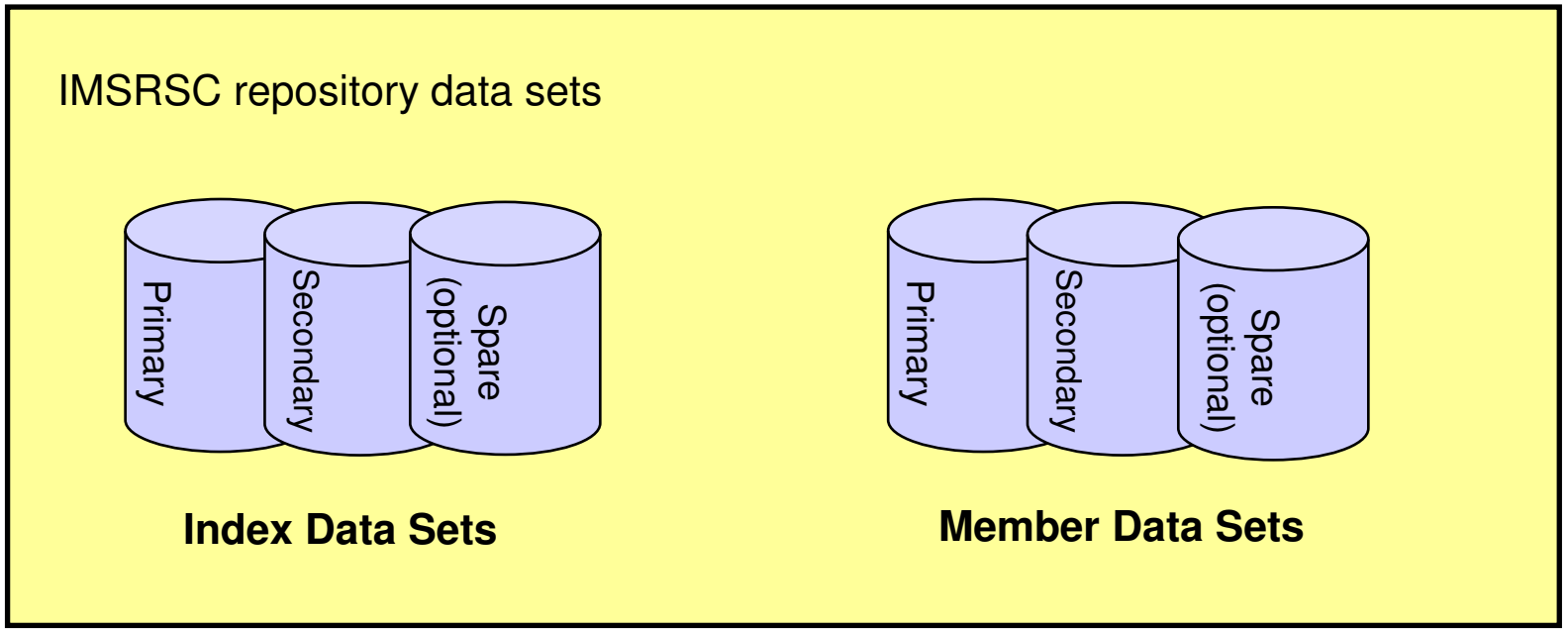


IMS Repository Function Components …

- IMS repositories
  - IMSRSC repository contains
    - Stored resource definitions for DRD resources for one or more DRD-enabled IMS systems
      - Programs / transactions / databases / FP routing codes and descriptors
    - Resource lists for each IMS
      - Contains resource names and resource types that can be processed by an IMS system
    - Changed resource lists for each IMS
      - Contain resource changes made when an IMS is down
  - Typically one IMSRSC repository per Repository Server (RS) per IMSplex

IMS Repository Function Components …

- IMS repositories
  - IMSRSC repository
    - Composed of up to three pairs of data sets
      - Primary index data set and primary member data set **(required)**
      - Secondary index data set and secondary member data set **(required)**
      - Spare index data set and spare member data set **(optional)**



IMS Repository Function Components …

- A Common Service Layer (CSL) IMSplex configuration consisting of
  - Operations Manager (OM)
    - Used for new/modified type-2 commands for repository functions
  - Resource Manager (RM)
    - Used for managing the new Repository Server (RS) address space
    - All online access to Repository Server is through RM address space
    - New type-2 commands for managing the Repository Server
      - UPDATE RM
      - QUERY RM
    - RM is enabled to the repository by specifying a Repository Section in the RM initialization member (CSLRIxxx)

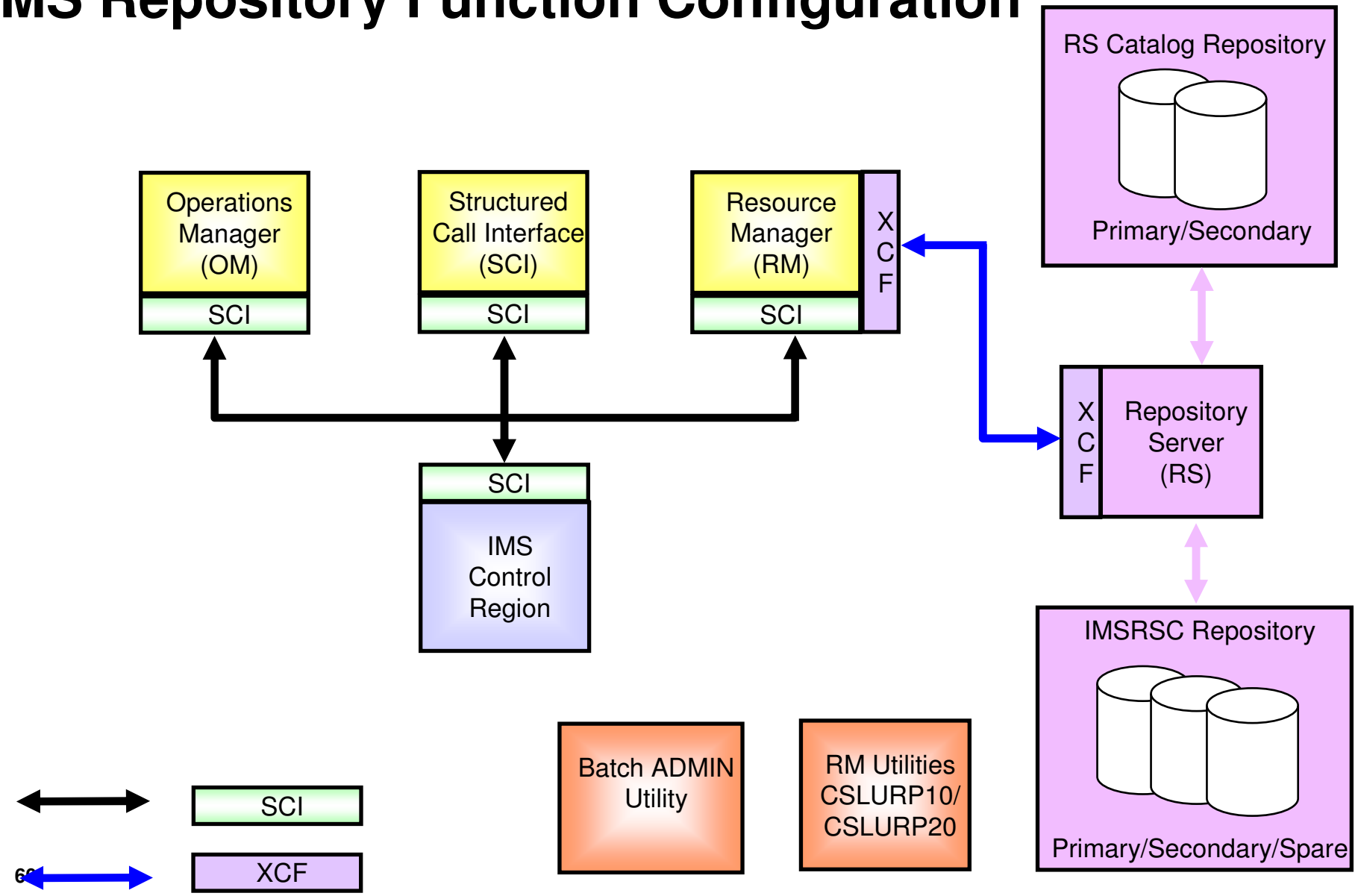
IMS Repository Function Components …

- A Common Service Layer (CSL) IMSplex configuration consisting of
  - Structured Call Interface (SCI)
    - Used for communications within the CSL
    - Not used for communications between RM and the RS
    - RS is not considered a CSL manager
  - Optionally, a resource structure in a Coupling Facility
    - Used for repository name and repository type consistency if present
    - Managed by a Common Queue Server (CQS) address space
    - Multiple RMs in an IMSplex require that a resource structure exists
  - SPOC (single point of control) for entering type-2 commands
  - Can be a single-IMS IMSplex or a multiple-IMS IMSplex

IMS Repository Function Components

- Batch utilities
  - Batch ADMIN utility (FRPBATCH)
    - Commands for managing IMSRSC repositories
      - Functions such as ADD a new IMSRSC repository, LIST the characteristics of an IMSRSC repository, START or STOP an IMSRSC repository
  - RDDS to / from repository utilities (Batch RM utilities)
    - RDDS to Repository Utility (CSLURP10)
      - For migration
    - Repository to RDDS Utility (CSLURP20)
      - For fallback

IMS Repository Function Configuration



IMS Repository Setup

- Repository Server setup
- Creating / enabling an IMSRSC repository for DRD

IMS Repository Setup

- Repository Server
  - Create catalog repository data sets
  - Set up BPE configuration member
  - Set up FRPCONFG configuration member
    - Repository Server settings
    - Audit log definitions
    - Define security
  - Set up the CSL
    - RM needs new CSLRIxxx repository section
  - Set up IMS
    - DFSDFxxx needs new repository section
  - Start the master Repository Server
  - Start subordinate Repository Servers

Enabling an IMSRSC repository for DRD

- Create IMSRSC repository data sets
- Define security for IMS repository
- Define the IMSRSC repository to the Repository Server
- Start the IMSRSC repository
- Enable the IMSRSC repository to RM and IMS
- Populate the IMSRSC repository
  - If IMS is up, populate using the EXPORT DEFN TARGET(REPO) command to add DRD stored resource definitions to the IMSRSC repository
  - If IMS is down, use the batch RDDS to Repository Utility (CSLURP10) to populate the IMSRSC repository

IMSRSC Repository is Active/Populated

- Begin to use type-2 DRD repository commands that access/update stored resource definitions in the repository

```
EXPORT DEFN TARGET(REPO) TYPE(ALL) NAME(*)

IMPORT DEFN SOURCE(REPO) TYPE(DB) NAME(DBABC)
OPTION(UPDATE)

DELETE DEFN TARGET(REPO) TYPE(DB) NAME(DBXYZ)

QUERY IMS

QUERY DB/PGM/TRAN/RTC SHOW(DEFN)
```

IMS Repository Commands

- IMS and RM IMSplex commands issued from SPOC or Manage Resource panels
- Batch ADMIN commands
- Repository Server commands issued through z/OS modify interface

IMS and RM IMSplex Commands issued from SPOC or Manage Resources panels

- IMS type-2 commands
  - UPDATE IMS
  - QUERY IMS

  For management of repository and RDDS functions

  For status of IMS

  - EXPORT DEFN TARGET(REPO)
  - IMPORT DEFN SOURCE(REPO)
  - DELETE DEFN

  For working with DRD stored resource definitions in the repository

  - QUERY DB/DBDESC/PGM/PGMDESC/TRAN/TRANDESC/RTC/RTCDESC SHOW(DEFN)

  For displaying stored resource definitions in the repository and their attributes

IMS and RM IMSplex Commands issued from SPOC or Manage Resources panels

- RM type-2 commands
  - UPDATE RM
  - QUERY RM

  } For management of repository and RDDS functions

  For status of RM

- DRD commands (CREATE, UPDATE, DELETE) work with runtime definitions, not the stored resource definitions in the repository

Batch ADMIN commands (FRPBATCH)

- Commands for managing repositories (IMSRSC)
  - ADD
  - UPDATE
  - RENAME
  - DELETE

  Add a new repository definition, update an existing repository definition, rename an existing repository definition, remove an existing repository definition.

  - DSCHANGE

    Change data set disposition

  - LIST

    List repository information

  - START
  - STOP

    Start or stop a repository

Repository Server Commands Issued Through the z/OS Modify Interface

- Functions for managing a Repository Server (RS) and its repositories (IMSRSC)

• ADMIN	Administrative functions for IMSRSC repositories – change data set disposition, display data sets, start/stop repositories
• AUDIT	Dynamically turn auditing on or off
• SECURITY	Refresh in-storage RACF profile definitions
• SHUTDOWN	Shutdown Repository Server address space(s)
• STOP	Stop/shutdown Repository Server

IMS Repository Migration Overview …

- From DRD with RDDSs to DRD with Repository
- From no DRD to DRD with Repository

IMS Repository Migration Overview …

- From DRD with RDDSs to DRD with Repository
  - Create non-system RDDS that contains current definitions via EXPORT command or DRD utilities
  - Set up repository parameters in PROCLIB members FRPCONFG, BPECONFG, CSLRIxxx, DFSDFxxx
  - Create catalog repository data sets and IMSRSC repository data sets
  - Start the Repository Server address space
  - Use the batch ADMIN utility to define the IMSRSC data sets to the Repository Server
  - Run the batch RDDS to Repository utility to populate the repository
  - Cold start IMS with AUTOIMPORT specified

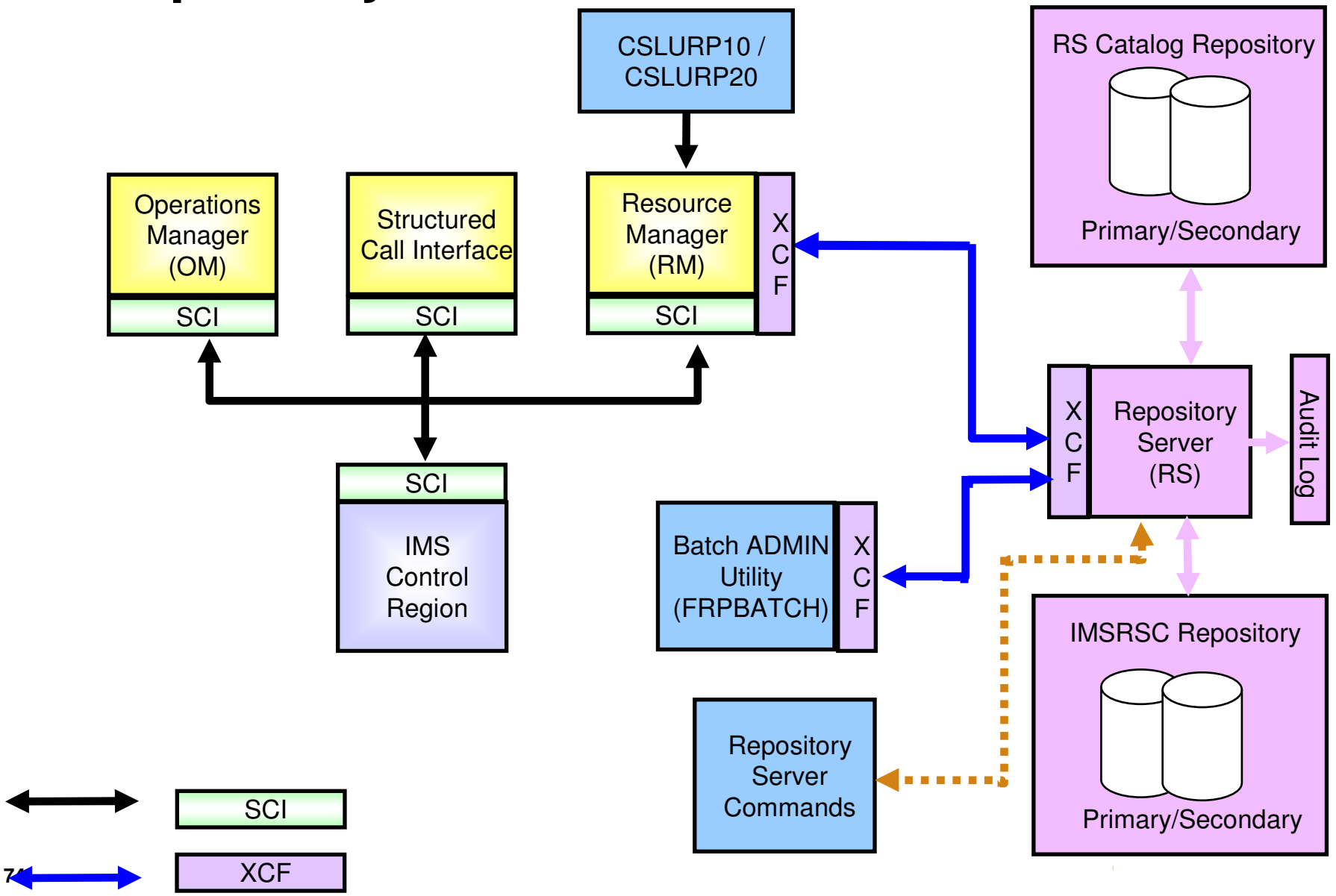
IMS Repository Migration Overview

- From no DRD to DRD with Repository
  - First implement DRD with RDDSs
    - Set up DRD parameters in DFSDFxxx
      - MODBLKS=DYN, RDDSDSN=, AUTOIMPORT=MODBLKS
    - Cold start IMS using updated DFSDFxxx
      - AUTOIMPORT will use MODBLKS for definitions
      - Online change now disabled
  - Follow process to migrate from DRD with RDDSs to DRD with repository

IMS Repository Function Benefits

- Consolidation of resource definitions in a single place, the repository
- DRD definitions are the initial implementation of the IMS repository function (to replace RDDSs)
- Full support for populating, managing, storing, sharing, and retrieving a consistent set of DRD stored resource definitions for multiple-IMS IMSplexes and single-IMS IMSplexes
- Manual coordination of multiple RDDSs in a multiple-IMS IMSplex eliminated, replaced by basic functioning of the IMS repository
- Improvements in IMSplex systems and resource management with the repository
- A strategic direction for IMS architecture

IMS Repository Function Architecture



Summary

- IMS repository function
- IMS repository function components
  - Repository Server (RS) address space
  - Repository catalog data sets
  - Repository data sets
  - CSL requirements and RM usage
- IMS repository setup
  - Repository Server setup
  - Creating/enabling an IMSRSC repository for DRD
- IMS repository commands and usage
- Migration to DRD with the repository