Dynamic Resources & The IMS Repository (IMSRSC) What, Why, and How

Ashley Lopez
Technical Enablement Specialist
IBM Z Ecosystem Infrastructure & WSC Advisory
ashlopez@ibm.com

Copyright© by SHARE Association Except where otherwise noted, this work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 3.0 license. http://creativecommons.org/licenses/by-nc-nd/3.0/

Agenda

WHAT

WHY

HOW

THE VALUE

IMS Repository – What is it?

- 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 single or multiple-IMS IMSplex to manage, store, share, and retrieve resource definitions
- **Focus:** improve the systems management and resource management aspects of handling IMS resource definitions
    - Across multiple IMSs or for a single standalone IMS

IMS Repository Function Usage – the Why

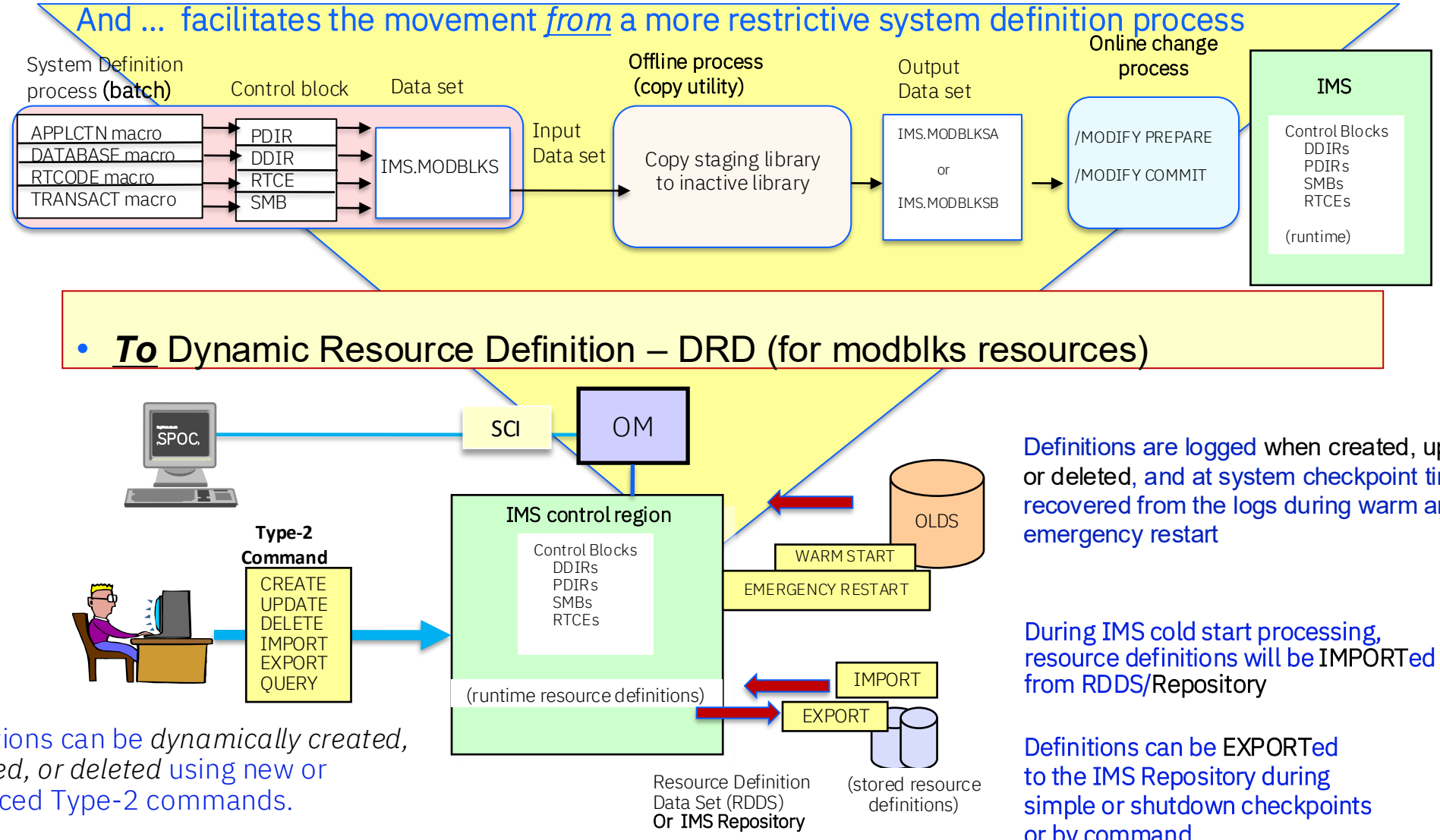
- Supports IMS DRD (Dynamic Resource Definition)
- Stores resource and descriptor definitions for Dynamic Resource Definition (DRD) can be stored in an IMS repository
  - Contains resource definitions for ***programs/transactions/databases/FP routing codes & descriptors***
    - Called the IMSRSC, the IMS resource definition repository
  - ***Provides an alternative to using MODBLKs with SYSGEN and online change***
  - **Considered a strategic alternative to the RDDS (BSAM dataset that was originally introduced to support 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
- IMS can retrieve the stored resource definitions from the IMSRSC repository to dynamically generate runtime resources during a COLD START

IMS Repository Function Usage – the Why

- Additionally
  - Changes made to active systems with the IMPORT SCOPE(ALL) command are applied to an inactive IMS when it is restarted
    - Resource lists for each IMS that uses the Repository
    - Support for generic definitions of a resource and specific attributes for an IMS if different
    - Change lists that contain resource and descriptor names that correspond to resource changes made for all the IMS systems with SCOPE(ALL) in a plex when an IMS is down
  - Simplifies IMSplex management
    - Easier to maintain synchronization of IMS systems' resources in a DRD with repository environment

Usage

With DRD –resource definition changes can be made to a running system and rebuilt across all restarts



Definitions can be *dynamically created, updated, or deleted* using new or enhanced Type-2 commands.

Definitions are logged when created, updated, or deleted, and at system checkpoint time and recovered from the logs during warm and emergency restart

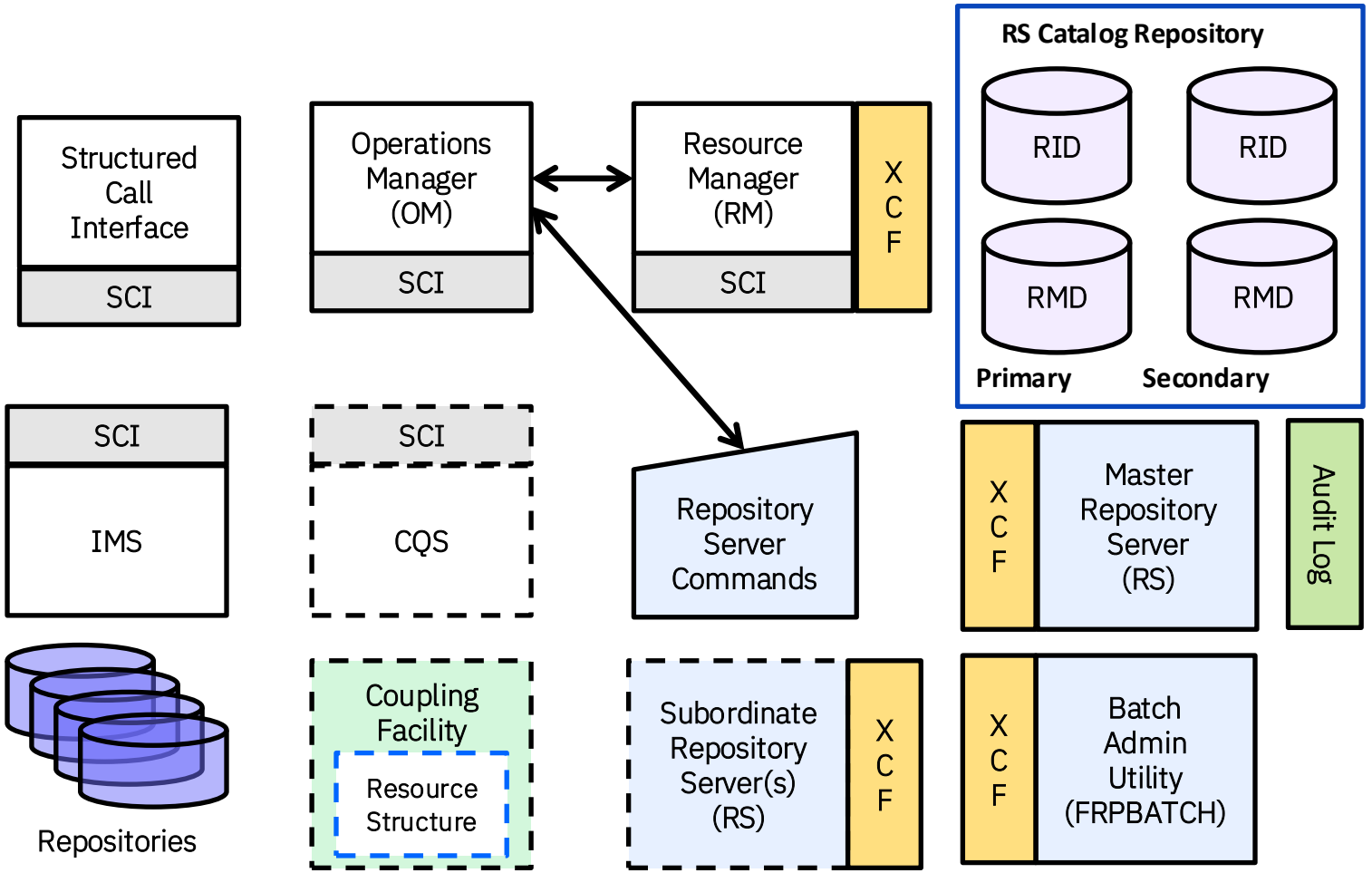
During IMS cold start processing, resource definitions will be IMPORTed from RDDS/Repository

Definitions can be EXPORTed to the IMS Repository during simple or shutdown checkpoints or by command

IMS Repository Function Components

- A Repository Server (RS)
  - A new BPE-based address space managed by the Resource Manager (RM) CSL address space
- Repositories
  - Catalog repository
    - Used by the Repository Server
  - IMSRSC repository(s)
    - Contains DRD stored resource definitions
- A 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 for the repository FRPBATCH
  - RDDS to Repository CSLURP10
  - Repository to RDDS CSKURP20 (for fallback)
- Audit Log (optional) – ability to write to the z/OS logger audit log stream

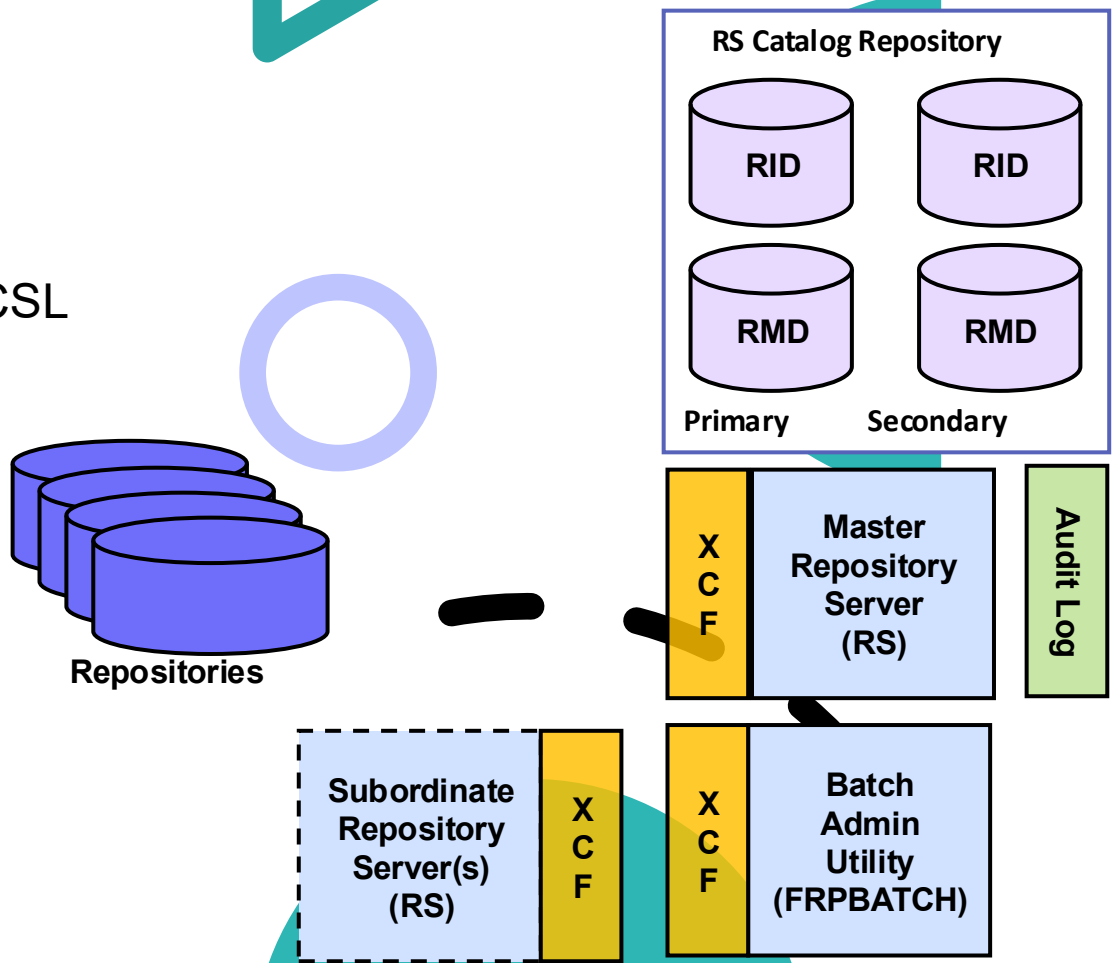
At a Glance


Structured Call Interface
SCI
Operations Manager (OM)
SCI
Resource Manager (RM)
SCI
XCF
RS Catalog Repository
RID
RID
RMD
RMD
Primary
Secondary
SCI
IMS
SCI
CQS
Repository Server Commands
XCF
Master Repository Server (RS)
Audit Log
Repositories
Coupling Facility
Resource Structure
Subordinate Repository Server(s) (RS)
XCF
XCF
Batch Admin Utility (FRPBATCH)

With a Little More Detail

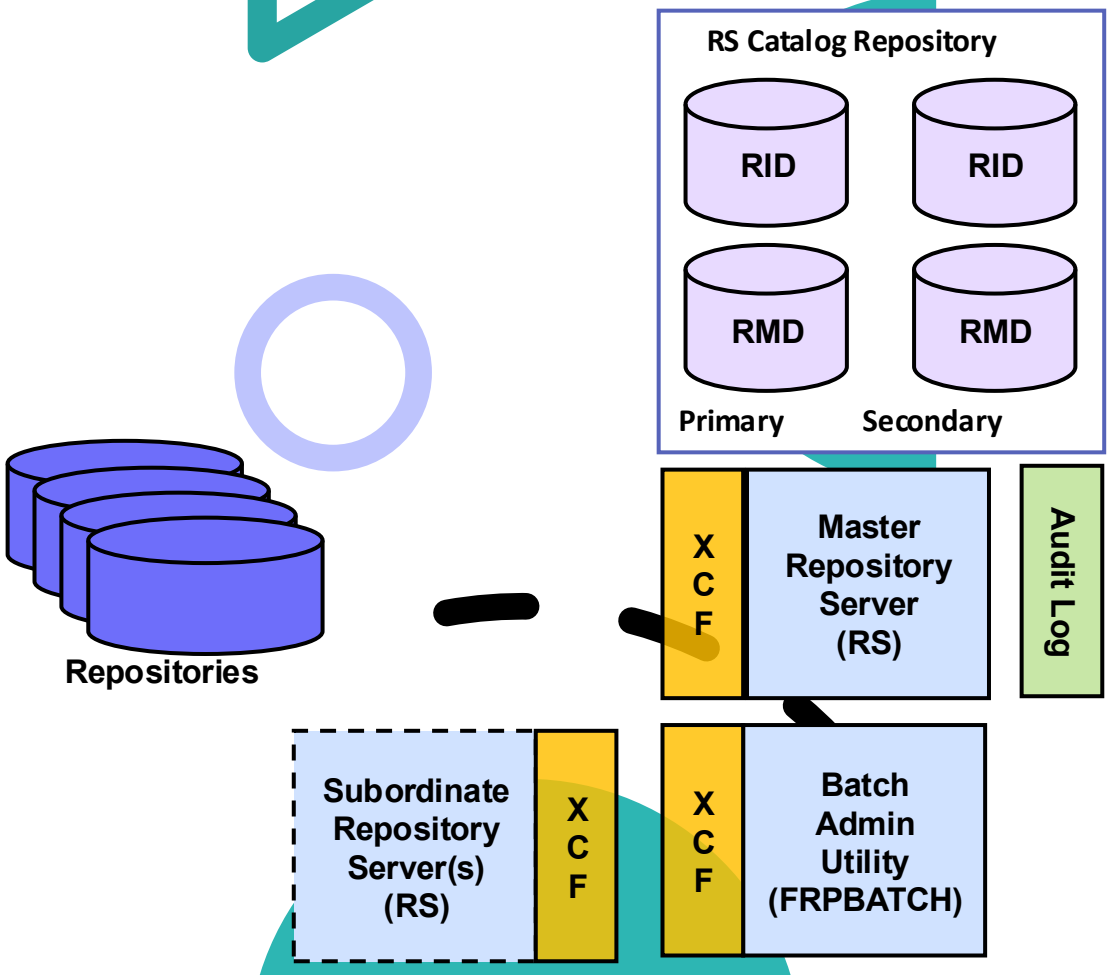
IMS Repository Function Components: Repository Server

- A Repository Server (RS)
  - A BPE-based address space
    - Managed by the Resource Manager (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***



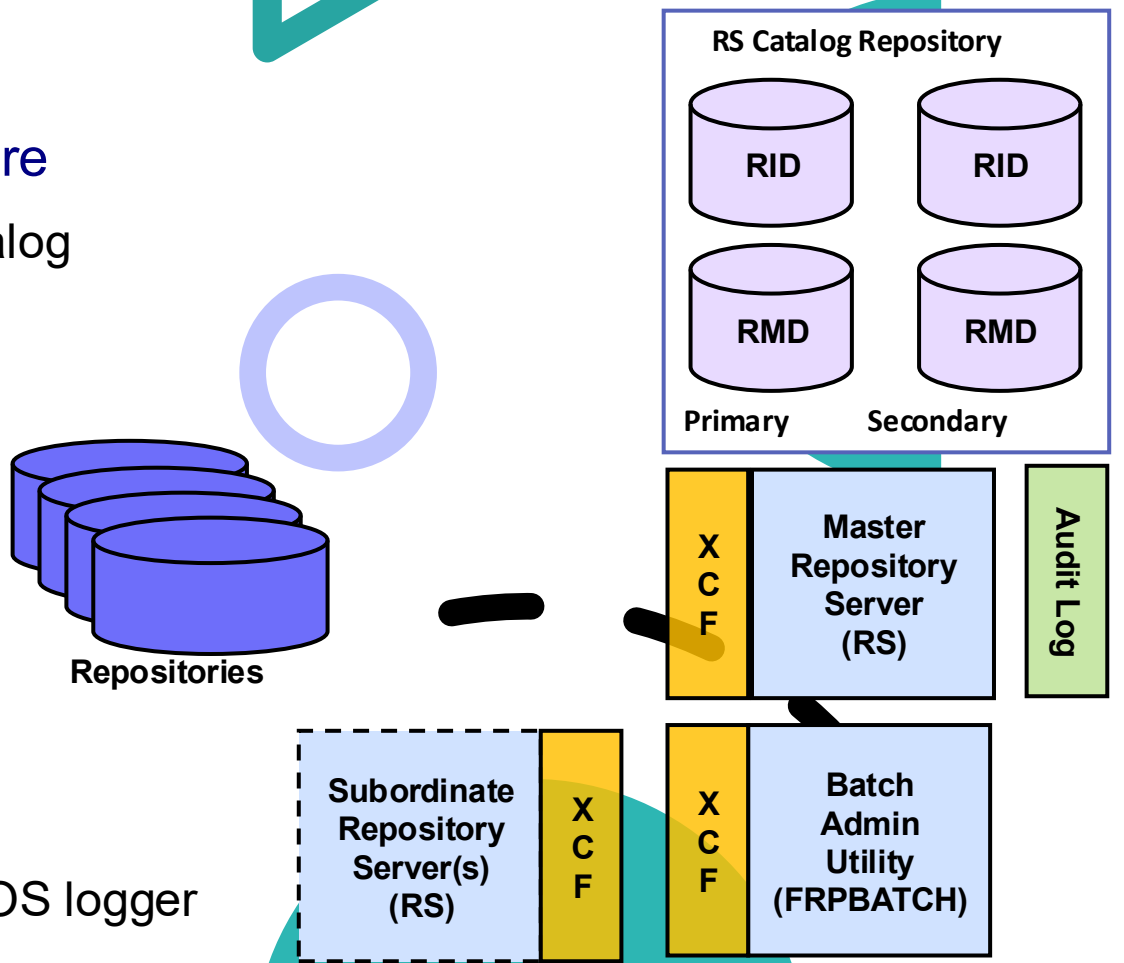
IMS Repository Function Components: Repository Server

- A Repository Server (RS)…
  - Uses VSAM KSDS data sets to store information
  - Leverages z/OS XCF for server communications
  - Can be on a separate LPAR in the parallel sysplex (XCF)
  - One master repository server per IMSplex
    - This master repository server can manage definitions from other IMSplexes in the sysplex (XCF) as well as its own IMSplex within the same repository
  - Recommendation: one master repository server address space per IMSplex

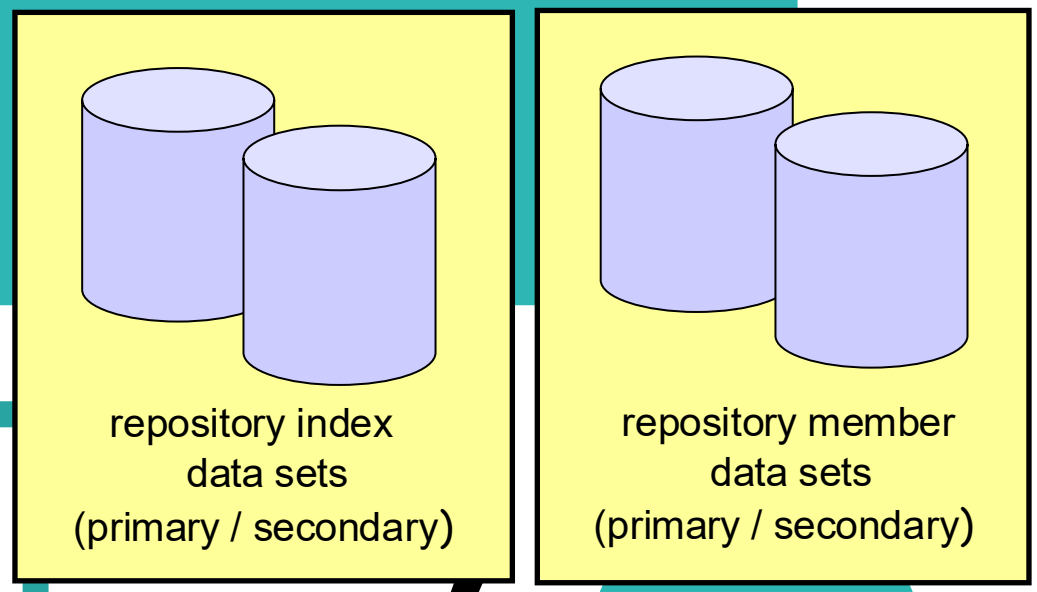


IMS Repository Function Components: Repository Server

- A Repository Server (RS) – Internal Structure
  - Has its own internal repository called the 'catalog repository'
  - Manages IMS repositories
    - IMS Catalog repository
    - IMS Resource repository (IMSRSC)
  - Manages registrations and connections to the repository
  - Ensures repository data integrity
  - Uses SAF to restrict access to repositories
  - Optionally, provides an audit trail using the z/OS logger
  - Provides tracing capabilities via BPE

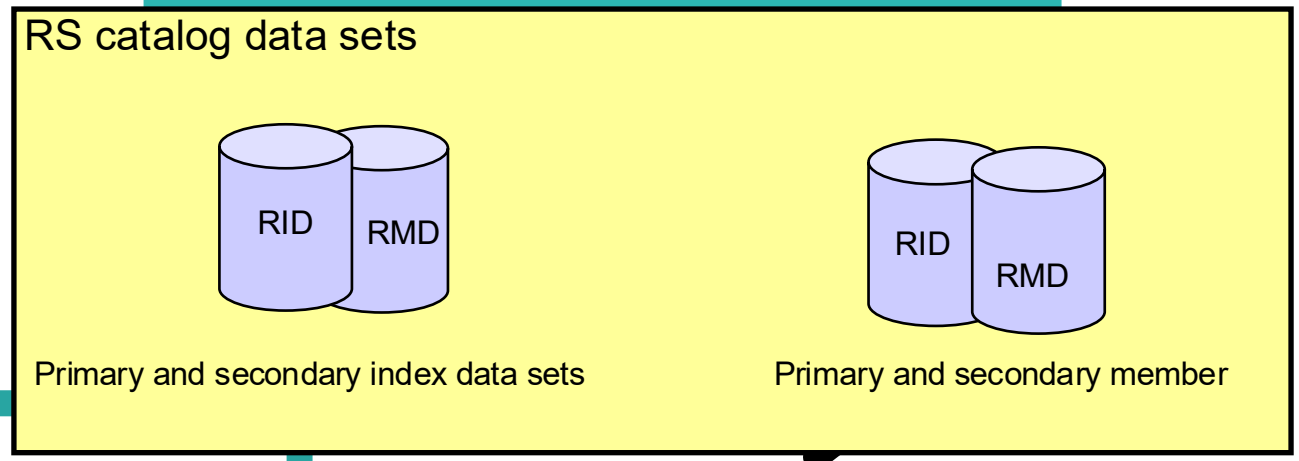


IMS Repository Function Components: Repository Data Sets



- 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
    - **Catalog repository** data sets
    - **IMS repository** data sets

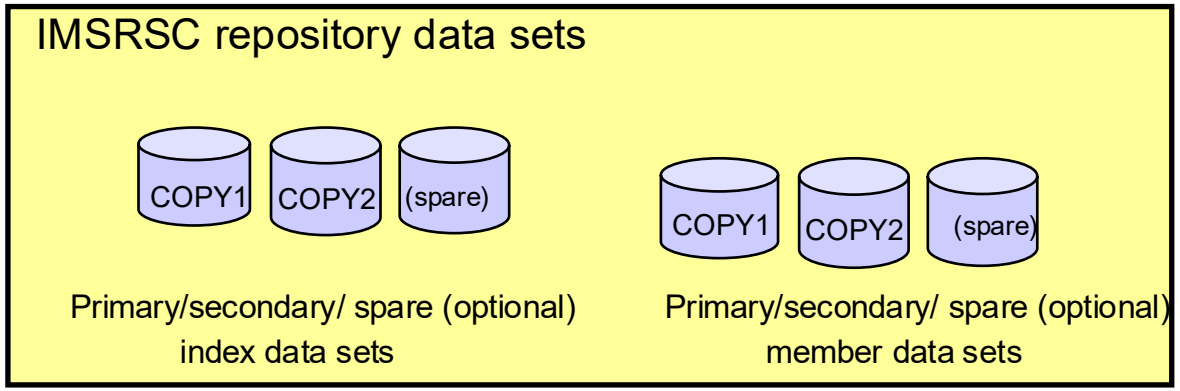
IMS Repository Function Components: RS Catalog Data Sets



- 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)
  - Each pair includes
    - Repository Index Data Set (RID)
      - Stores names and keys of Repository members
    - Repository Member Data Set (RMD)
      - Associates Repository names with Repository data sets

IMS Repository Function components – IMSRSC Repository

- IMS Resource (IMSRSC) repository
  - Contains the stored resource definitions for DRD resources for one or more DRD-enabled IMS systems
    - Resource lists for each IMS
    - Stored resource definitions (programs/transactions/databases/FP routing codes)



- Composed of up to three pairs of data sets
  - Primary index data set and primary member data set (required) (COPY1)
  - Secondary index data set and secondary member data set (required) (COPY2)
  - Spare index data set and spare member data set (optional and recommended) (SPARE)

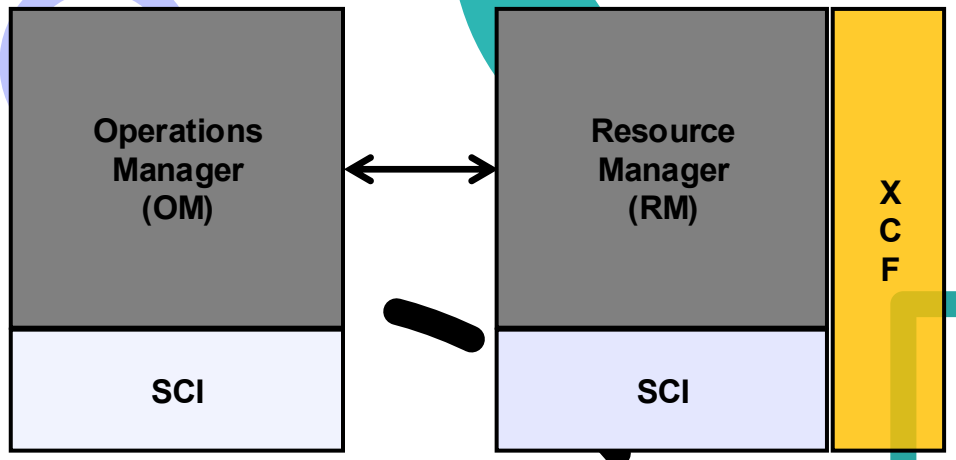
- ***States*** of these data sets are maintained in RS catalog and can be:
  - Primary (COPY1), Secondary (COPY2), Spare (SPARE), Discarded (DISCARD)
    - Note: IMSRSC repository must have valid primary/secondary sets of data sets or it is stopped

IMS Repository Function components – IMSRSC Repository…

- IMSRSC repository – Why include a Spare pair?
  - Spare recovery capability
    - Works at ‘pair’ level
    - Empty data sets used on write error on primary or secondary pair
      - Read error attempts read to other good data set
    - Repository server performs recovery automatically if a spare pair is available
    - Valid data set pair (either primary or secondary) is copied to spare pair
    - Failed pair marked ‘discarded’
    - IMSRSC repository stopped during this recovery
    - IMSRSC repository restarted with new primary/secondary copies
    - User must define a new pair of empty spare data sets, then use the batch ADMIN utility or repository server commands to set their status to ‘spare’

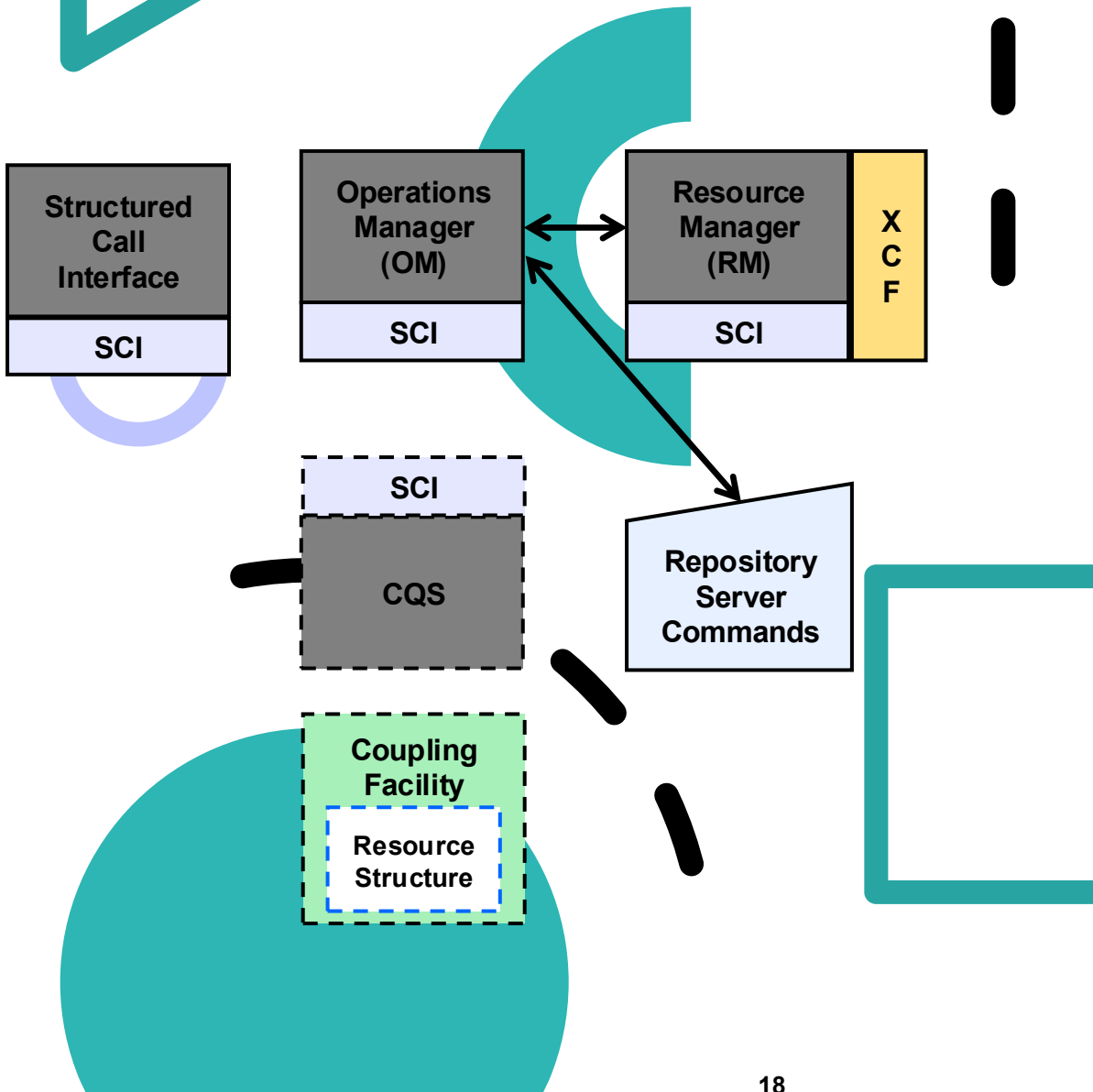
The Common Service Layer (CSL) Environment & The Repository

- The use of the IMS Repository requires 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 via XCF Services
    - New type-2 commands for enabling/managing repository server
      - UPDATE RM
      - QUERY RM
    - RM registers to the repository server and connects to the IMSRSC repository(s) during RM initialization
    - RM is enabled to the repository by specifying a Repository Section in the RM initialization member (CSLRIxxx) or via the UPDATE RM command



CSL & The Repository

- 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 (uses XCF)
    - RS is not considered a CSL manager
    - Repository server can optionally register with SCI (shows as part of the IMSplex on QUERY IMSPLEX output) though repository server communications are through RM with XCF
  - 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



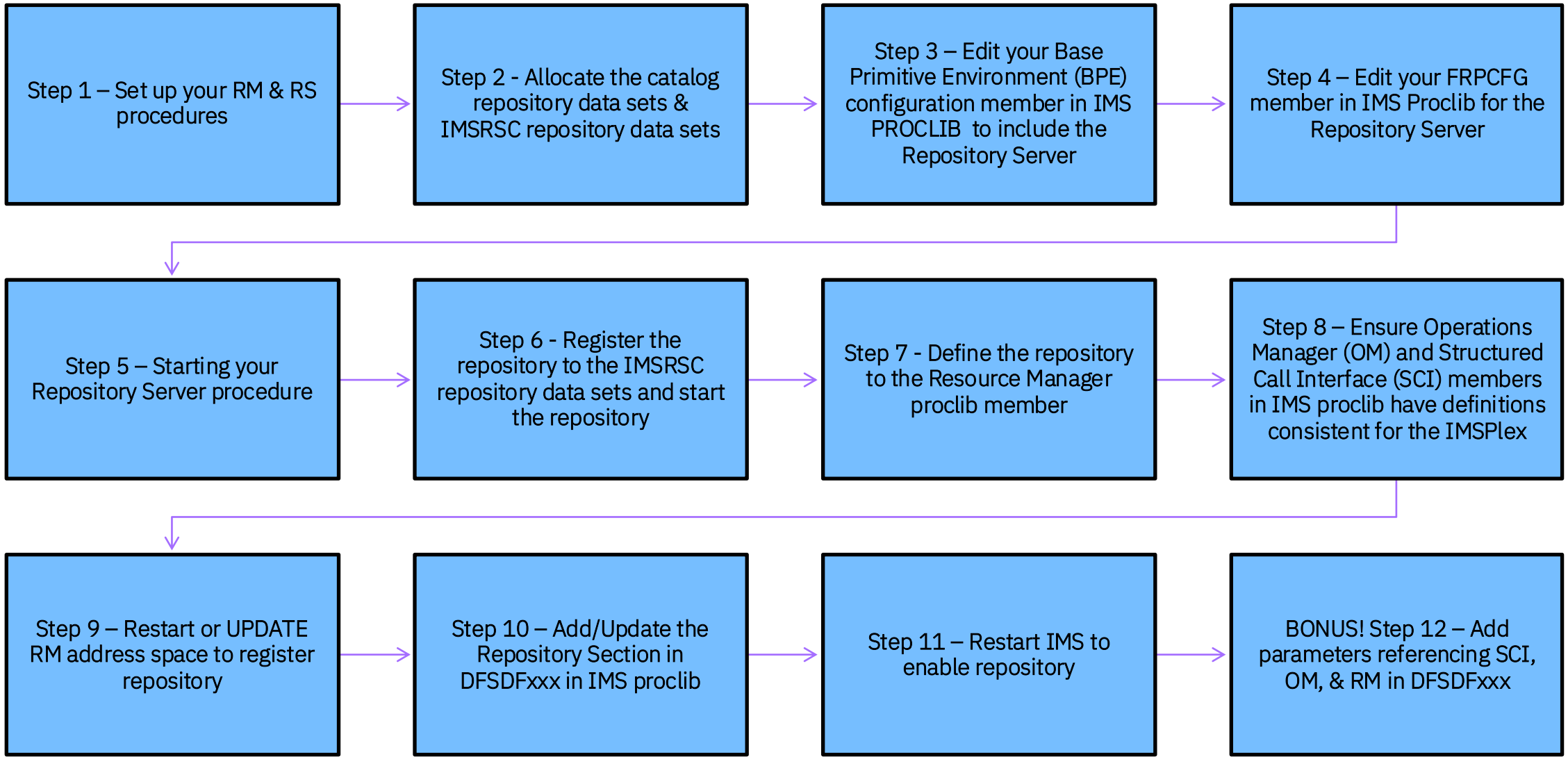
Security

- https://ibm.biz/BdapaJ
  - *https://www.ibm.com/docs/en/ims/15.6.0?topic=environments-restricting-access-rs-catalog-repository-imsrsc-repository*
- Basic steps:
  - Set up via SAF profiles for RM access to a particular repository
    - Called ‘Connection Security’
  - Set up via SAF profiles for caller access to RM
    - Authorized callers, like IMS
    - Non-authorized callers, like the batch RM utilities
      - ‘Member level security’ needs to be set up
  - Set up SAF profiles for access to the catalog repository data sets
  - Set up SAF profiles for access to IMSRSC repository
  - Set up SAF profiles for access to members in an IMSRSC repository
  - Set up SAF profiles for who can make changes to the AUDIT level settings
  - Security for batch access via USERID on JOB statement

IMS Repository Set-up

- https://www.ibm.com/docs/en/ims/15.6.0?topic=repository-defining-imsrsc

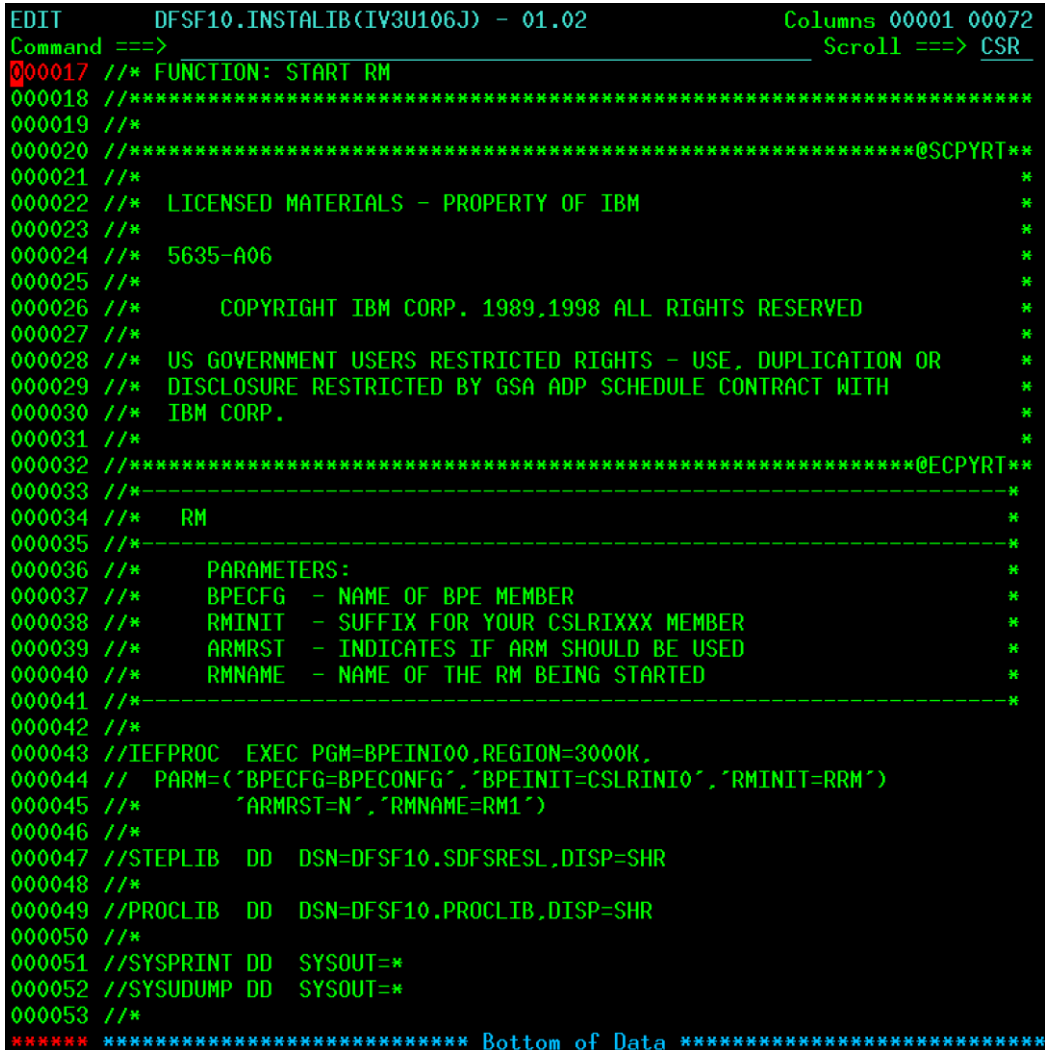
My Implementation Steps for the IMS Repository Server



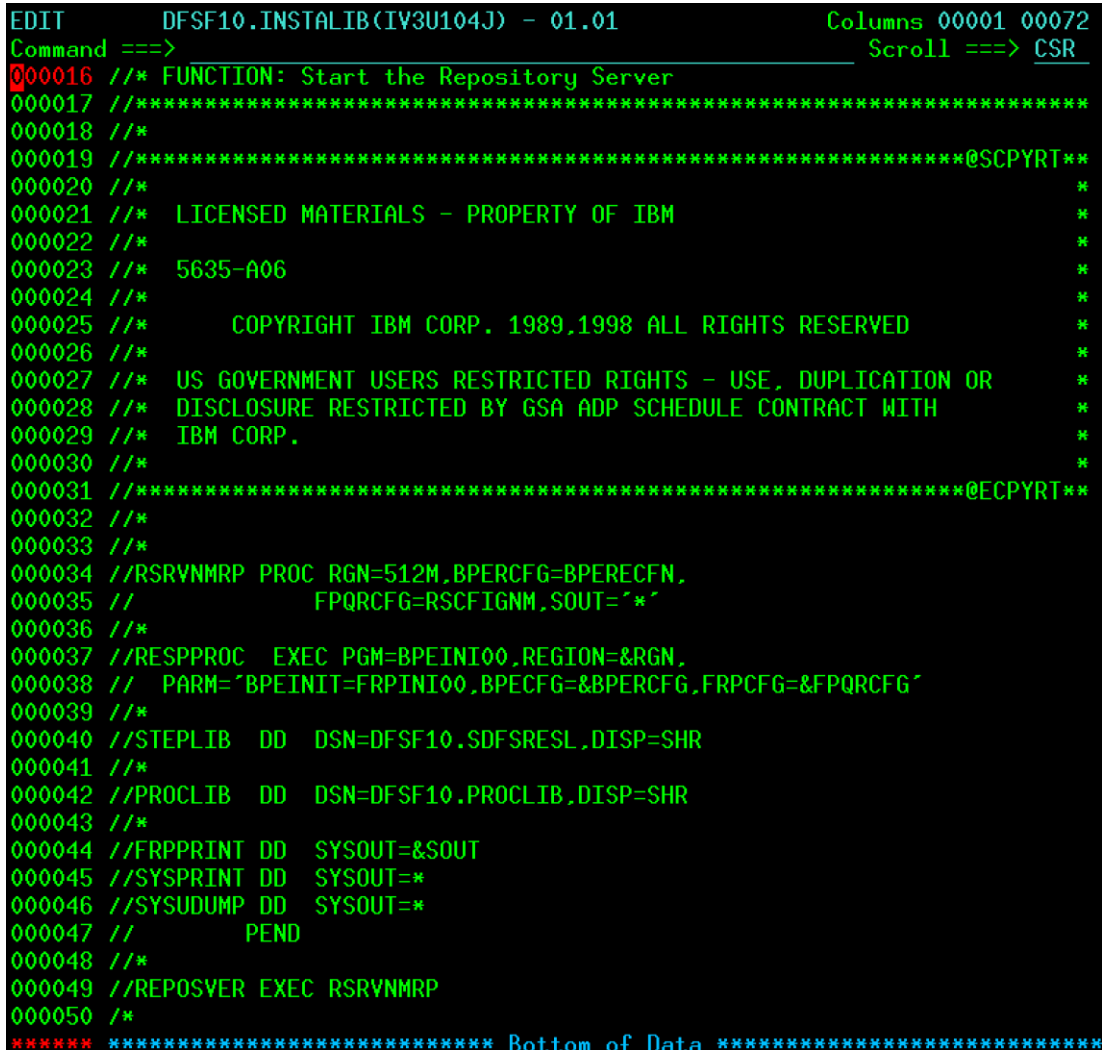
Step 1 – Set up your RM & RS procedures

Find sample procedures from the IVPs in the IMS INSTALIB data set and copy into the system proclib

- Resource Manager (RM) - IV3U106J
- Repository Server (RS) - IV3U104J



```
EDIT       DFSF10.INSTALIB(IV3U106J) - 01.02            Columns 00001 00072
Command ===>                                             Scroll ===> CSR
000017 //* FUNCTION: START RM
000018 //*********************************************************************
000019 //*
000020 //*************************************************************@SCPYRT**
000021 //*                                                                    *
000022 //*  LICENSED MATERIALS - PROPERTY OF IBM                              *
000023 //*                                                                    *
000024 //*  5635-A06                                                          *
000025 //*                                                                    *
000026 //*      COPYRIGHT IBM CORP. 1989,1998 ALL RIGHTS RESERVED             *
000027 //*                                                                    *
000028 //*  US GOVERNMENT USERS RESTRICTED RIGHTS - USE, DUPLICATION OR        *
000029 //*  DISCLOSURE RESTRICTED BY GSA ADP SCHEDULE CONTRACT WITH           *
000030 //*  IBM CORP.                                                         *
000031 //*                                                                    *
000032 //*************************************************************@ECPYRT**
000033 //*--------------------------------------------------------------------*
000034 //*   RM                                                               *
000035 //*--------------------------------------------------------------------*
000036 //*     PARAMETERS:                                                    *
000037 //*     BPECFG  - NAME OF BPE MEMBER                                   *
000038 //*     RMINIT  - SUFFIX FOR YOUR CSLRIXXX MEMBER                      *
000039 //*     ARMRST  - INDICATES IF ARM SHOULD BE USED                      *
000040 //*     RMNAME  - NAME OF THE RM BEING STARTED                         *
000041 //*--------------------------------------------------------------------*
000042 //*
000043 //IEFPROC  EXEC PGM=BPEINI00,REGION=3000K,
000044 //  PARM=('BPECFG=BPECONFG','BPEINIT=CSLRINI0','RMINIT=RRM')
000045 //*       'ARMRST=N','RMNAME=RM1')
000046 //*
000047 //STEPLIB  DD  DSN=DFSF10.SDFSRESL,DISP=SHR
000048 //*
000049 //PROCLIB  DD  DSN=DFSF10.PROCLIB,DISP=SHR
000050 //*
000051 //SYSPRINT DD  SYSOUT=*
000052 //SYSUDUMP DD  SYSOUT=*
000053 //*
****** ***************************** Bottom of Data ****************************
```



```
EDIT       DFSF10.INSTALIB(IV3U104J) - 01.01            Columns 00001 00072
Command ===>                                             Scroll ===> CSR
000016 //* FUNCTION: Start the Repository Server
000017 //*********************************************************************
000018 //*
000019 //*************************************************************@SCPYRT**
000020 //*                                                                    *
000021 //*  LICENSED MATERIALS - PROPERTY OF IBM                              *
000022 //*                                                                    *
000023 //*  5635-A06                                                          *
000024 //*                                                                    *
000025 //*      COPYRIGHT IBM CORP. 1989,1998 ALL RIGHTS RESERVED             *
000026 //*                                                                    *
000027 //*  US GOVERNMENT USERS RESTRICTED RIGHTS - USE, DUPLICATION OR        *
000028 //*  DISCLOSURE RESTRICTED BY GSA ADP SCHEDULE CONTRACT WITH           *
000029 //*  IBM CORP.                                                         *
000030 //*                                                                    *
000031 //*************************************************************@ECPYRT**
000032 //*
000033 //*
000034 //RSRVNMRP PROC RGN=512M,BPERCFG=BPERECFN,
000035 //            FPQRCFG=RSCFIGNM,SOUT='*'
000036 //*
000037 //RESPPROC  EXEC PGM=BPEINI00,REGION=&RGN,
000038 //  PARM='BPEINIT=FRPINI00,BPECFG=&BPERCFG,FRPCFG=&FPQRCFG'
000039 //*
000040 //STEPLIB  DD  DSN=DFSF10.SDFSRESL,DISP=SHR
000041 //*
000042 //PROCLIB  DD  DSN=DFSF10.PROCLIB,DISP=SHR
000043 //*
000044 //FRPPRINT DD  SYSOUT=&SOUT
000045 //SYSPRINT DD  SYSOUT=*
000046 //SYSUDUMP DD  SYSOUT=*
000047 //        PEND
000048 //*
000049 //REPOSVER EXEC RSRVNMRP
000050 /*
****** ***************************** Bottom of Data ****************************
```

Step 1 (cont.) – Set up your RM & RS procedures

Adjusted the system steplib & proclib concatenations for my environment
Referenced the IMS proclib definitions for the Repository Server and Resource Manager

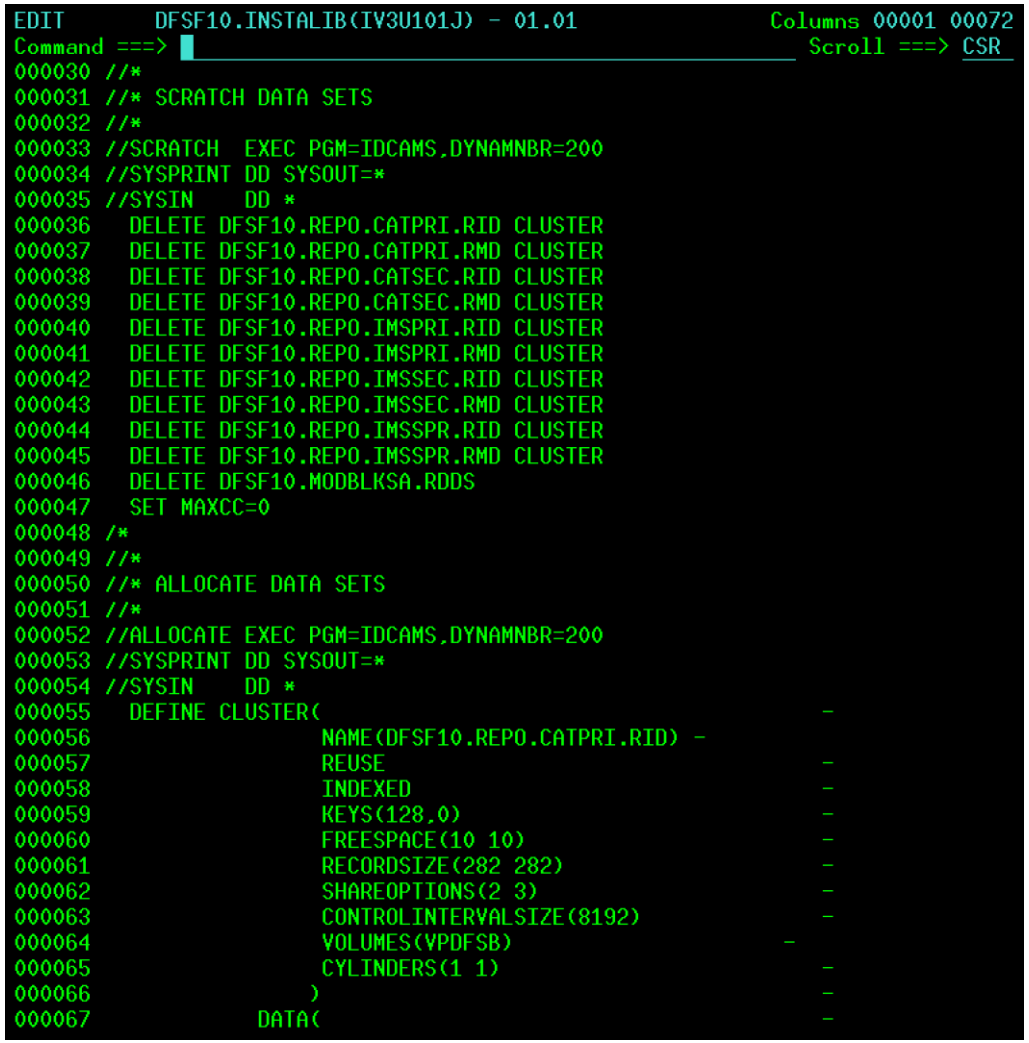
```
EDIT       VENDOR.PROCLIB(IMS15RM1) - 01.00              Columns 00001 00072
Command ===> _____________________________________________ Scroll ===> CSR
000031 //*
000032 //IEFPROC  EXEC PGM=BPEINI00,REGION=3000K,
000033 //  PARM=('BPECFG=BPECONFG','BPEINIT=CSLRINI0','RMINIT=000',
000034 //         'ARMRST=N','RMNAME=RM1')
000035 //*
000036 //STEPLIB  DD  DSN=DFSF10.USER.SDFSRESL,DISP=SHR
000037 //         DD  DSN=DFSF10.SDFSRESL,DISP=SHR
000038 //PROCLIB  DD DSN=VENDOR.PROCLIB,DISP=SHR
000039 //         DD DSN=SVTSC.PROCLIB,DISP=SHR
000040 //         DD DSN=DFSF10.PROCLIB,DISP=SHR
000041 //*
000042 //SYSPRINT DD  SYSOUT=*
000043 //SYSUDUMP DD  SYSOUT=*
000044 //*
****** ***************************** Bottom of Data ****
```

```
EDIT       VENDOR.PROCLIB(IMS15RS1) - 01.00              Columns 00001 00072
Command ===> _____________________________________________ Scroll ===> CSR
002300 //*
002400 //RSRVNMR  PROC RGN=512M,BPERCFG=BPERECFN,
002500 //             FPQRCFG=RSCFIGNM,SOUT='*'
002600 //*
002700 //RESPPROC  EXEC PGM=BPEINI00,REGION=&RGN,
002800 //  PARM='BPEINIT=FRPINI00,BPECFG=&BPERCFG,FRPCFG=&FPQRCFG'
002900 //*
003000 //STEPLIB  DD  DSN=DFSF10.USER.SDFSRESL,DISP=SHR
003100 //         DD  DSN=DFSF10.SDFSRESL,DISP=SHR
003200 //*
003300 //PROCLIB  DD DSN=VENDOR.PROCLIB,DISP=SHR
003400 //         DD DSN=SVTSC.PROCLIB,DISP=SHR
003500 //         DD DSN=DFSF10.PROCLIB,DISP=SHR
003600 //*
003700 //FRPPRINT DD  SYSOUT=&SOUT
003800 //SYSPRINT DD  SYSOUT=*
003900 //SYSUDUMP DD  SYSOUT=*
004000 //*
004100 //*REPOSVER EXEC RSRVNMRP
004200 //*
```

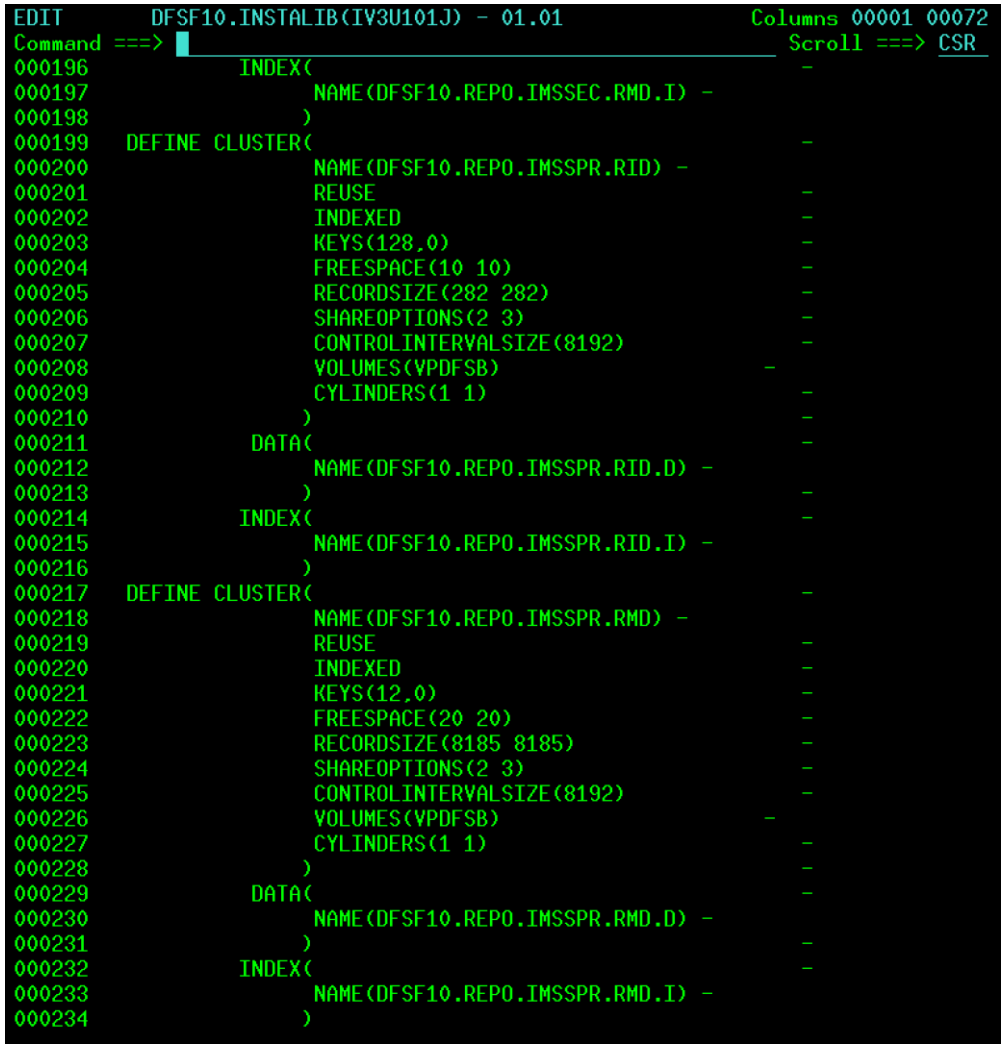
Step 2 - Allocate the RS catalog repository data sets & IMSRSC repository data sets

Find sample JCL member, IV3U101J, from the IVPs in the IMS INSTALIB data set

- The job creates two sets of each plus a spare – KSDS data sets



```
EDIT       DFSF10.INSTALIB(IV3U101J) - 01.01            Columns 00001 00072
Command ===>                                             Scroll ===> CSR
000030 //*
000031 //* SCRATCH DATA SETS
000032 //*
000033 //SCRATCH  EXEC PGM=IDCAMS,DYNAMNBR=200
000034 //SYSPRINT DD SYSOUT=*
000035 //SYSIN    DD *
000036   DELETE DFSF10.REPO.CATPRI.RID CLUSTER
000037   DELETE DFSF10.REPO.CATPRI.RMD CLUSTER
000038   DELETE DFSF10.REPO.CATSEC.RID CLUSTER
000039   DELETE DFSF10.REPO.CATSEC.RMD CLUSTER
000040   DELETE DFSF10.REPO.IMSPRI.RID CLUSTER
000041   DELETE DFSF10.REPO.IMSPRI.RMD CLUSTER
000042   DELETE DFSF10.REPO.IMSSEC.RID CLUSTER
000043   DELETE DFSF10.REPO.IMSSEC.RMD CLUSTER
000044   DELETE DFSF10.REPO.IMSSPR.RID CLUSTER
000045   DELETE DFSF10.REPO.IMSSPR.RMD CLUSTER
000046   DELETE DFSF10.MODBLKSA.RDDS
000047   SET MAXCC=0
000048 /*
000049 //*
000050 //* ALLOCATE DATA SETS
000051 //*
000052 //ALLOCATE EXEC PGM=IDCAMS,DYNAMNBR=200
000053 //SYSPRINT DD SYSOUT=*
000054 //SYSIN    DD *
000055   DEFINE CLUSTER(                                -
000056                  NAME(DFSF10.REPO.CATPRI.RID) -
000057                  REUSE                          -
000058                  INDEXED                        -
000059                  KEYS(128,0)                    -
000060                  FREESPACE(10 10)               -
000061                  RECORDSIZE(282 282)            -
000062                  SHAREOPTIONS(2 3)              -
000063                  CONTROLINTERVALSIZE(8192)      -
000064                  VOLUMES(VPDFSB)                -
000065                  CYLINDERS(1 1)                 -
000066                 )                               -
000067            DATA(                                -
```



```
EDIT       DFSF10.INSTALIB(IV3U101J) - 01.01            Columns 00001 00072
Command ===>                                             Scroll ===> CSR
000196            INDEX(                               -
000197                 NAME(DFSF10.REPO.IMSSEC.RMD.I) -
000198                )
000199   DEFINE CLUSTER(                                -
000200                 NAME(DFSF10.REPO.IMSSPR.RID) -
000201                 REUSE                          -
000202                 INDEXED                        -
000203                 KEYS(128,0)                    -
000204                 FREESPACE(10 10)               -
000205                 RECORDSIZE(282 282)            -
000206                 SHAREOPTIONS(2 3)              -
000207                 CONTROLINTERVALSIZE(8192)      -
000208                 VOLUMES(VPDFSB)                -
000209                 CYLINDERS(1 1)                 -
000210                )                               -
000211             DATA(                              -
000212                 NAME(DFSF10.REPO.IMSSPR.RID.D) -
000213                )                               -
000214            INDEX(                              -
000215                 NAME(DFSF10.REPO.IMSSPR.RID.I) -
000216                )
000217   DEFINE CLUSTER(                                -
000218                 NAME(DFSF10.REPO.IMSSPR.RMD) -
000219                 REUSE                          -
000220                 INDEXED                        -
000221                 KEYS(12,0)                     -
000222                 FREESPACE(20 20)               -
000223                 RECORDSIZE(8185 8185)          -
000224                 SHAREOPTIONS(2 3)              -
000225                 CONTROLINTERVALSIZE(8192)      -
000226                 VOLUMES(VPDFSB)                -
000227                 CYLINDERS(1 1)                 -
000228                )                               -
000229             DATA(                              -
000230                 NAME(DFSF10.REPO.IMSSPR.RMD.D) -
000231                )                               -
000232            INDEX(                              -
000233                 NAME(DFSF10.REPO.IMSSPR.RMD.I) -
000234                )
```

Step 3 – Edit your Base Primitive Environment (BPE) configuration member in IMS PROCLIB to include the Repository Server

Tracing definitions for capturing internal events within the Repository Server
Take the defaults as shown or adjust as necessary for your environment's requirements

```
EDIT       DFSF10.PROCLIB(BPERECFN) - 01.00                Columns 00001 00072
Command ===>                                                 Scroll ===> CSR
****** ******************************** Top of Data ********************************
000001 # DEFINITIONS FOR REPO TRACES
000002 TRCLEV=(*,HIGH,REPO,PAGES=300)/* DEFAULT ALL TRACES TO HIGH */
****** ***************************** Bottom of Data ******************************
```

Step 4 – Edit your FRPCFG member in IMS Proclib for the Repository Server

- Defines RS configuration parameters for performance, communications, and security
- Also identifies the names of the catalog repository data sets to be used
- Always keep in mind the RSNAME, IMSPLEX, XCF Group Name definitions

```
EDIT       DFSF10.PROCLIB(RSCFIGNM) - 01.00
Command ===>
****** ***************************** Top of Data ****
000001 XCF_THREADS=8
000002 MAX_COMMUNICATION_RETRY=32
000003 MBR_CORE_MAX=1024
000004 IMSPLEX(NAME=PLEX1)
000005 RSNAME=RSRVNM
000006 PRIMARY_CATALOG_REPOSITORY_INDEX=(
000007 DFSF10.REPO.CATPRI.RID)
000008 PRIMARY_CATALOG_REPOSITORY_MEMBER=(
000009 DFSF10.REPO.CATPRI.RMD)
000010 SECONDARY_CATALOG_REPOSITORY_INDEX=(
000011 DFSF10.REPO.CATSEC.RID)
000012 SECONDARY_CATALOG_REPOSITORY_MEMBER=(
000013 DFSF10.REPO.CATSEC.RMD)
000014 VSAM_BUFNO=256
000015 VSAM_BUFSIZE=8
000016 XCF_GROUP_NAME=REPXCFGN
000017 AUDIT=NO
000018 AUDIT_FAIL=CONTINUE
000019 AUDIT_LEVEL=HIGH
000020 AUDIT_DEFAULT=NOAUDIT
****** ***************************** Bottom of Data **
```

Step 5 – Starting your Repository Server procedure

```
STC01183 ---- WEDNESDAY, 06 NOV 2024 ----
STC01183  IEF695I START IMS15RS1 WITH JOBNAME IMS15RS1 IS ASSIGNED TO USER STCOP
STC01183  $HASP373 IMS15RS1 STARTED
STC01183  FRP2006I - Server starting: Release 1.2.0, XCF group REPXCFGN RSRVNMRP
STC01183  FRP2012I - Opening repository: CATALOG
STC01183  FRP2016I - Repository opened: CATALOG
STC01183  FRP2002I - Master repository server status obtained RSRVNMRP
STC01183  FRP2025I - Server start completed RSRVNMRP
STC01183  FRP2026I - XCF group REPXCFGN joined successfully RSRVNMRP
STC01183  FRP2012I - Opening repository: RSRVNM
STC01183  IEC161I 152-061,IMS15RS1,IMS15RS1,SYS00009,,,DFSF10.REPO.IMSSPR.RID,
          IEC161I DFSF10.REPO.IMSSPR.RID.D,CATALOG.DFSF10
STC01183  IEC161I 152-061,IMS15RS1,IMS15RS1,SYS00010,,,DFSF10.REPO.IMSSPR.RMD,
          IEC161I DFSF10.REPO.IMSSPR.RMD.D,CATALOG.DFSF10
STC01183  FRP2016I - Repository opened: RSRVNM
```

/S **IMS15RS1** to start the RS address space

```
IEC161I 152-061,IMS15RS1,IMS15RS1,SYS00009,,,DFSF10.REPO.IMSSPR.RID,
IEC161I DFSF10.REPO.IMSSPR.RID.D,CATALOG.DFSF10
IEF237I 1A54 ALLOCATED TO SYS00010
IEC161I 152-061,IMS15RS1,IMS15RS1,SYS00010,,,DFSF10.REPO.IMSSPR.RMD,
IEC161I DFSF10.REPO.IMSSPR.RMD.D,CATALOG.DFSF10
FRP2016I - Repository opened: RSRVNM                                  RSRV
The following configuration parameters are in effect:
   Number of XCF message threads (XCF_THREADS)              8
   XCF group name (XCF_GROUP_NAME)                          REPXCFGN
   Max storage before allocating data space (MBR_CORE_MAX)  1024K
   SAF security class name (SAF_CLASS)
   Number of VSAM buffers to use VSAM I/O (VSAM_BUFNO)      256
   VSAM buffer size (VSAM_BUFSIZE)                          8K
   Server-busy max retry attempts (MAX_COMMUNICATION_RETRY) 32
   Catalog primary RID data set name                        DFSF10.REPO.CATPRI.
   Catalog primary RMD data set name                        DFSF10.REPO.CATPRI.
   Catalog secondary RID data set name                      DFSF10.REPO.CATSEC.
   Catalog secondary RMD data set name                      DFSF10.REPO.CATSEC.
   Audit enabled? (AUDIT)                                   NO
******************************** BOTTOM OF DATA ********************************
```

Step 6 - Register the repository to the IMSRSC repository data sets and start the repository

ADD and START FRPBATCH commands

Manually send commands to the repository

Submit JCL job – Sample batch job found in IVP dataset, **IV3U105J**

```
EDIT       DFSF10.INSTALIB(IV3U105J) - 01.01              Columns 00001 00072
Command ===>                                               Scroll ===> CSR
000033 //*
000034 //REPOADD  EXEC PGM=FRPBATCH,PARM='XCFGROUP=REPXCFGN'
000035 //STEPLIB  DD DISP=SHR,DSN=DFSF10.SDFSRESL
000036 //SYSPRINT DD SYSOUT=*
000037 //SYSIN    DD *
000038  ADD REPOSITORY(IMSREPNM)                          +
000039      REPDSN1RID(DFSF10.REPO.IMSPRI.RID)    +
000040      REPDSN1RMD(DFSF10.REPO.IMSPRI.RMD)    +
000041      REPDSN2RID(DFSF10.REPO.IMSSEC.RID)    +
000042      REPDSN2RMD(DFSF10.REPO.IMSSEC.RMD)    +
000043      REPDSN3RID(DFSF10.REPO.IMSSPR.RID)    +
000044      REPDSN3RMD(DFSF10.REPO.IMSSPR.RMD)    +
000045      AUTOOPEN(YES)
000046  START REPOSITORY(IMSREPNM)
000047 //*
****** ******************************** Bottom of Data ********************************
```

RSNAME definition goes here

Step 7 - Define the repository to the Resource Manager proclib member

```
EDIT       DFSF10.PROCLIB(CSLRI000) - 01.00                Columns 00001 00072
Command ===>                                                Scroll ===> CSR
****** ********************************* Top of Data ********************************
000001 *-------------------------------------------------------------------*
000002 * RM INITIALIZATION PROCLIB MEMBER.                                 *
000003 *-------------------------------------------------------------------*
000004 ARMRST=N,                        /* SHOULD ARM RESTART RM ON FAILURE  */
000005 RMNAME=RM1,                      /* RM NAME (RMID = RM1RM)            */
000006 IMSPLEX(NAME=PLEX1)              /* IMSPLEX NAME (CSLPLEX1)           */
000007 *-------------------------------------------------------------------*
000008 * REPOSITORY SECTION ADDED 10/02/24                                 *
00000                                                              -------*
00001    <SECTION=REPOSITORY>
00001    * REPOSITORY=(NAME=RSRVNMRP,TYPE=IMSRSC,GROUP=REPXCFGN)
00001    REPOSITORY=(NAME=RSRVNM,TYPE=IMSRSC,GROUP=REPXCFGN)
00001    * AUDITACCESS=NOAUDIT | SECURITY | UPDATE | READ| SYSTEMREAD
00001                                                              -------*
000015 * END OF MEMBER CSLRI000                                           *
000016 *-------------------------------------------------------------------*
******* ***************************** Bottom of Data ***************************
```

Step 8 – Ensure Operations Manager (OM) and Structured Call Interface (SCI) members in IMS proclib has definitions consistent for the IMSPlex

```
EDIT       DFSF10.PROCLIB(CSLOI000) - 01.00            Columns 00001 00072
Command ===>                                            Scroll ===> CSR
****** ***************************** Top of Data ********************************
000001 *-------------------------------------------------------------------*
000002 * OM INITIALIZATION PROCLIB MEMBER - CSLOI000                        *
000003 *-------------------------------------------------------------------*
000004 ARMRST=N,                    /* SHOULD ARM RESTART OM ON FAILURE  */
000005 CMDLANG=ENU,                 /* USE ENGLISH FOR COMMMAND DESC     */
000006 CMDSEC=N,                    /* NO COMMAND SECURITY               */
000007 OMNAME=OM1,                  /* OM NAME (OMID = OM1OM)            */
000008 IMSPLEX(
000009   NAME=PLEX1,                /* IMSPLEX NAME (CSLPLEX1)           */
000010   AUDITLOG=SYSLOG.OM2Q01.LOG), /* MVS LOG STREAM                  */
000011 CMDTEXTDSN=DFSF10.SDFSDATA /* CMD TEXT DATASET          */
000012 *-------------------------------------------------------------------*
000013 * END OF MEMBER CSLOI000
000014 *----------------------------
****** ***************************** B
```

```
EDIT       DFSF10.PROCLIB(CSLSI000) - 01.00            Columns 00001 00072
Command ===>                                            Scroll ===> CSR
****** ***************************** Top of Data ********************************
000001 *-------------------------------------------------------------------*
000002 * SCI INITIALIZATION PROCLIB MEMBER.                                 *
000003 *-------------------------------------------------------------------*
000004 ARMRST=N,                    /* SHOULD ARM RESTART SCI ON FAILURE  */
000005 SCINAME=SCI1,                /* SCI NAME (SCIID = SCI1SCI)         */
000006 IMSPLEX(NAME=PLEX1)          /* IMSPLEX NAME (CSLPLEX1)            */
000007 *-------------------------------------------------------------------*
000008 * END OF MEMBER CSLSI000                                            *
000009 *-------------------------------------------------------------------*
****** ***************************** Bottom of Data *****************************
```

Step 9 – Restart or UPDATE RM address space to register repository

Submit Type-2 Command: **UPDATE RMTYPE(REPO) REPOTYPE(IMSRSC)**
Manual PURGE and START of the Resource Manager address space in SDSF
Either option will pick up the new Repository Server definitions

Look for the message *CSL2502A RM WAITING FOR MASTER REPOSITORY SERVER... to confirm that the Repository Server is registered

Step 10 – Add/Update the Repository Section in DFSDFxxx in IMS proclib

```
EDIT       DFSF10.PROCLIB(DFSDF000) - 01.09                Columns 00001 00072
Command ===> _____________________________________________ Scroll ===> CSR
****** ******************************** Top of Data ********************************
000001 *-------------------------------------------------------------------*
000002 * REPOSITORY SECTION - ADDED 10/01/24                               *
000003 *-------------------------------------------------------------------*
000004 <SECTION=REPOSITORY>
000005 REPOSITORY=(TYPE=(IMSRSC))
000006 *-------------------------------------------------------------------*
```

Step 11 – Restart IMS to enable repository

New startup process:
IRLM Region → /S IMS15RL1
RS Address Space → /S IMS15RS1
Control Region → /S IMS15CR1
SCI address space → IMS15SCI
OM Address Space → /S IMS15OM1
RM Address Space → /S IMS15RM1
Control Region with a COLD START → /S IMS15CR1… /R ##,/NRE CHKPT 0 FORMAT ALL

Note: IMS READY prompt from WTO will appear after RS initialization is complete.

New Shutdown:
Checkpoint → /R ##,/CHE SNAPQ
Terminate Control Region → /R ##,/CHE FREEZE
RS Address Space → /P IMS15RS1
RM Address Space → /P IMS15RM1
OM Address Space → /P IMS15OM1
SCI Address space → /P IMS15SCI
IRLM Region → /P IMS15RL1

BONUS! Step 12 – Add parameters referencing SCI, OM, & RM in DFSDFxxx

Automatically starting Common Service Layer address spaces

This won't start Repository Server procedure but brings out the 3 address spaces required when the Control Region is initiated

The Repository Server should be the address space that comes up after RM has initiated

To bring down the CSL address spaces during shutdown, manually purge each region OR issue one of the two following commands :

- /F scijobname,SHUTDOWN CSLLCL
- /F scijobname,SHUTDOWN CSLPLEX

```
*-------------------------------------------------------------*
* COMMON SERVICE LAYER SECTION                                *
*-------------------------------------------------------------*
<SECTION=COMMON_SERVICE_LAYER>
CMDSEC=N                        /* NO CMD AUTHORIZATION CHECKING   */
IMSPLEX=PLEX1                   /* IMSPLEX NAME                    */
MODBLKS=DYN                     /* DRD ENABLED:MODBLKS OLC DISABLE */
OMPROC=IMS15OM1                 /* OPERATIONS MANAGER              */
RMENV=Y                         /* RESOURCE MANAGER ENABLED        */
RMPROC=IMS15RM1                 /* RESOURCE MANANGER PROC          */
SCIPROC=IMS15SCI                /* SERVICE CALL INTERFACE          */
*-------------------------------------------------------------*
```

Step 12 (cont.) – Restart IMS to enable automatic startup of CSL

New startup process:
IRLM Region → /S IMS15RL1
RS Address Space → /S IMS15RS1
Control Region → /S IMS15CR1
(SCI, OM, & RM address spaces will initialize in succession automatically)
COLD START → /R ##,/NRE CHKPT 0 FORMAT ALL

Note: IMS READY prompt from WTO will appear after RS initialization is complete.

New Shutdown:
Checkpoint → /R ##,/CHE SNAPQ
Terminate Control Region → /R ##,/CHE FREEZE
CSL Termination → /F IMS15SCI,SHUTDOWN CSLLCL or CSLPLEX
(SCI, OM, & RM address spaces will terminate in succession automatically)
RS Address Space → /P IMS15RS1
IRLM Region → /P IMS15RL1

IMS Repository Value

Experience this in action through TSO SPOC!
EXEC ‘imshlq.SDFSEXEC(DFSAPPL)' 'HLQ(imshlq)'

```
PLEX1                    IMS Single Point of Control
Command ===>

                          Plex . . PLEX1  Route . . IVP1       Wait . .
 Response for: QUERY TRAN NAME(IVTNO) SHOW(ALL)                More:    >
Trancode MbrName    CC LPSBname LCls    LQCnt  LLCT LPLCT LPLCTTime LCPRI LNPRI
IVTNO    IVP1        0 DFSIVP1     1        0 65535 65535   6553500     1     1
```

```
PLEX1                    IMS Single Point of Control
Command ===>

                          Plex . . PLEX1  Route . . IVP1       Wait . .
 Response for: QUERY TRAN NAME(IVTNXX) SHOW(ALL)               More:    >
Trancode MbrName    CC CCText              LPSBname LCls    LQCnt  LLCT LPLCT LPL
IVTNXX   IVP1       10 NO RESOURCES FOUND
```

QUERY existing and nonexistent resources like transactions

Experience this in action though TSO SPOC!

```
                        IMS Single Point of Control
 Command ===>

                          Plex . . PLEX1  Route . . IVP1       Wait . .
 Response for: CREATE TRAN NAME(IVTNXX) SET(PGM(DFSIVP1) CLASS(...
Trancode MbrName    CC
IVTNXX   IVP1        0
```

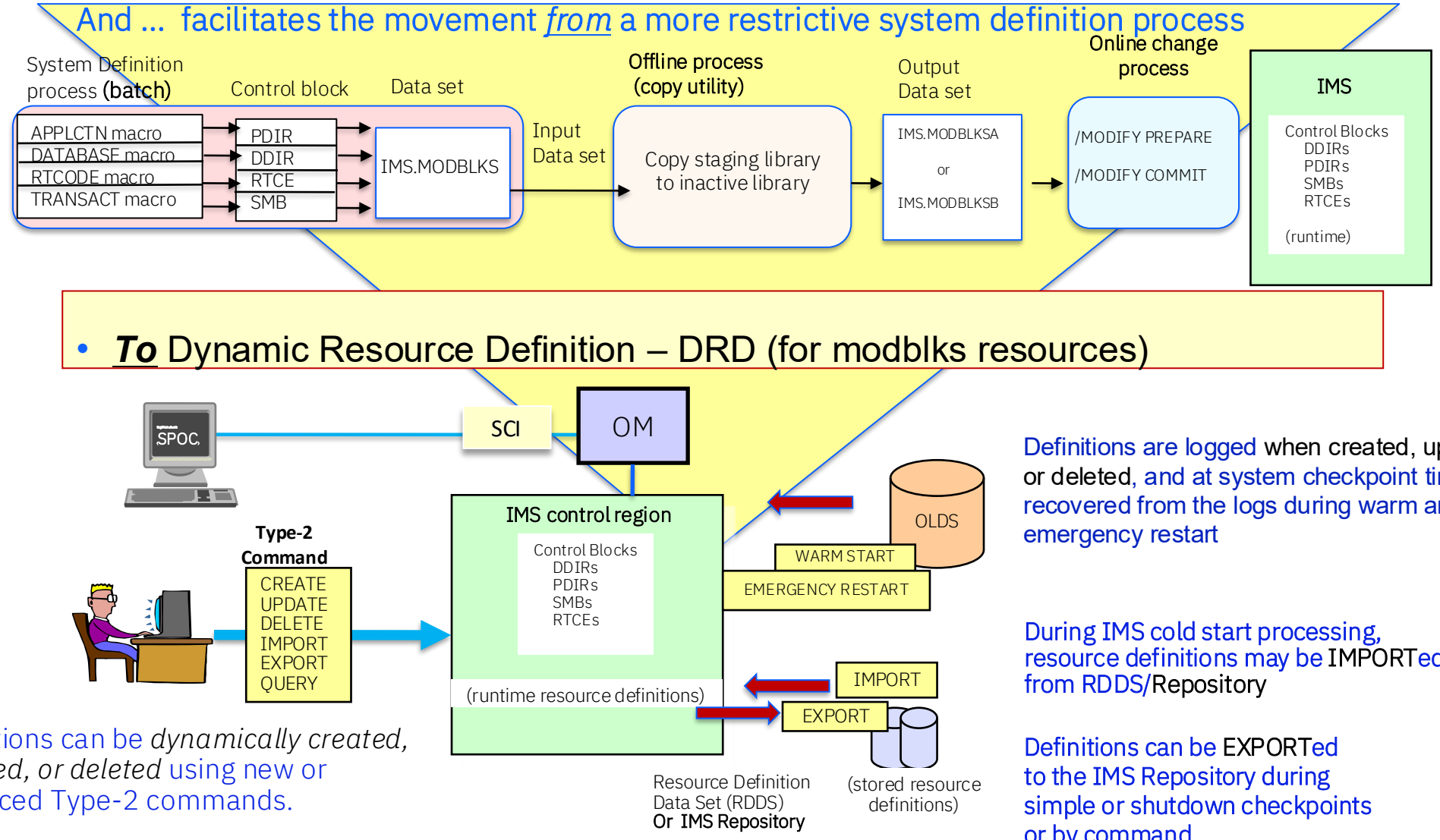
```
                        IMS Single Point of Control
 Command ===>

                          Plex . . PLEX1  Route . . IVP1       Wait . .
 Response for: QUERY TRAN NAME(IVTNXX) SHOW(ALL)                More:   >
Trancode MbrName    CC LPSBname LCls    LQCnt  LLCT LPLCT LPLCTTime LCPRI LNPRI
IVTNXX   IVP1        0 DFSIVP1     3        0 65535 65535   6553500     1     1
```

CREATE a new transaction, and then QUERY it to see it already registered!

Usage

With DRD –resource definition changes can be made to a running system and rebuilt across all restarts



Definitions can be *dynamically created, updated, or deleted* using new or enhanced Type-2 commands.

Definitions are logged when created, updated, or deleted, and at system checkpoint time and recovered from the logs during warm and emergency restart

During IMS cold start processing, resource definitions may be IMPORTed from RDDS/Repository

Definitions can be EXPORTed to the IMS Repository during simple or shutdown checkpoints or by command

Value of the IMS Repository for DRD

- 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 in a single place
- Provides improved availability
  - Repository can be enabled/disabled without an IMS outage via command
  - Duplexing of data plus spare capability improves data availability
- Provides single source consistency for DRD stored resource definitions
  - No need for multiple sets of RDDSs in a multiple-IMS IMSplex
  - No need for coordinating multiple sets of RDDSs in a multiple-IMS IMSplex
  - Repository architecture controls consistency and integrity of data

A strategic IMS architectural direction
Based upon BPE, CSL, IMSplex architecture

Questions?
Comments?

Thank you!

Your feedback is important!

SHARE
EDUCATE • NETWORK • INFLUENCE

Submit a session evaluation for each session you attend:

www.share.org/evaluation

Copyright© by SHARE Association Except where otherwise noted, this work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 3.0 license. http://creativecommons.org/licenses/by-nc-nd/3.0/

Want to attend an in-person IBM z/OS Academy?

z/OS ACADEMY

Learn, **Interact** and **Network** with IBMers and peers

May 5th- 7th, 2026	**Fall 2026**
IBM Tech Campus	**IBM US**
Ehningen, Germany	**New York, USA**

These **free** events are designed for early tenure z/OS system programmers (2-10 years), but all are welcome!

Training and presentations include topics on new z/OS capabilities, best practices, career tips, and **much more!**
Subscribe to the community page today to stay informed about future events!

Register now for Ehningen/ Germany:

Join our IBM Community: https://ibm.biz/zOSAcademy
Questions? Contact us at zOS.Academy.USA@us.ibm.com **or** zOS.Academy.Europe@de.ibm.com

Experience more with IBM

Visit us at the IBM Booth #113

After a full day of technical sessions, take a break with us!

Connect with our experts, snap a photo with the z17 Plexi or the latest Telum II, and get an up-close look at our Spyre Accelerator.

Come back each day for fresh topics and demos at our expert stations.

Think 2026

Join 5000+ senior business and technology leaders who are seizing the AI revolution to unlock unprecedented growth and productivity at Think 2026.

Find out more information using the QR code below.

IBM Digital Asset Haven

IBM Digital Asset Haven is the operational backbone for financial institutions and regulated enterprises entering the digital asset economy.

Find out more information using the QR code below.

IBM Corporation 2026