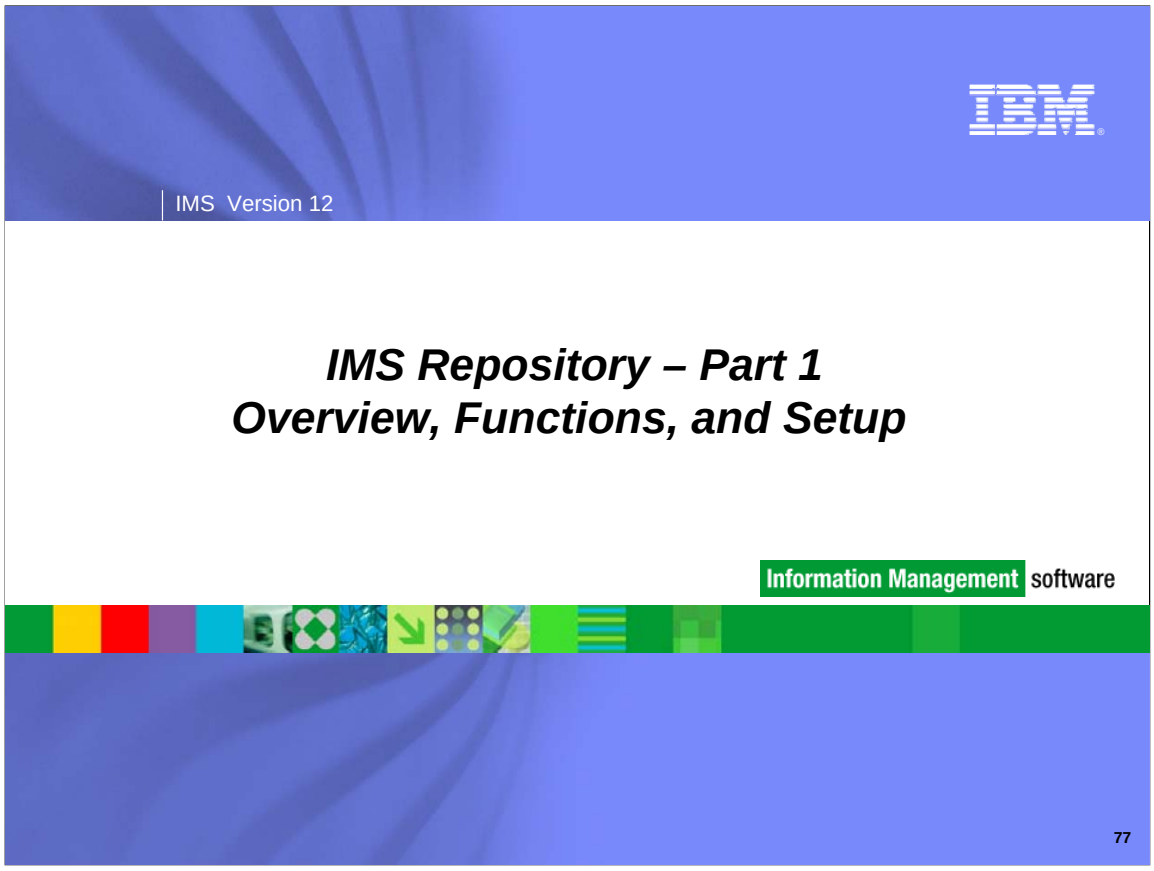


This topic covers the new IMS Repository function in IMS 12. There are two sections to this topic: the first covers an overview of the repository, the functions provided by the repository including new terminology, and how to set up the repository; the second covers how to use the repository including command usage, differences between using the repository for DRD vs. RDDSs for DRD, security considerations, and migration considerations.

IMS Repository – Part 1

- Overview of the IMS Repository
  - IMS Repository function and components
- IMS Repository Function Infrastructure
  - Repository Server (RS) address space
  - Repository catalog data sets
  - Repository data sets
  - CSL requirements and RM usage
- IMS Repository set-up
  - Repository Server setup
  - Creating an IMSRSC repository for DRD
- IMS Repository access
  - Online access through RM
  - Direct access without RM (batch)

78

Part 1 of the IMS Repository topic covers an overview of the repository and its major components, more details of the infrastructure and components of the repository, and set-up considerations. Methods of accessing the repository (online and batch) will also be discussed. These topics will describe ‘what the repository is’.

IMS Repository – Part 2

- IMS repository commands
  - IMS and RM IMSplex commands issued from SPOC
  - Batch ADMIN commands
  - Repository Server commands issued through z/OS modify interface
- Comparison of DRD use with RDDS versus repository
- Using DRD with the IMS repository in an online environment
- Managing the IMS repository in an offline batch environment
- Migration to repository
- Security considerations
- DRD user interface enhancements
- IVP enhancements for repository
- Summary

79

Part 2 of the IMS Repository topic covers commands, usage, and repository management. There are several types of commands that will be discussed, some comparisons of using DRD with RDDS vs. the repository, some repository usage considerations, followed by migration and security considerations. Related topics of the DRD user interface and the IVP will also be covered. These topics describe ‘how to use the repository’.

Overview of the IMS Repository

First let’s discuss the highlights of IMS Repository in IMS 12.

IMS Repository Function

- 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

81

Any repository implementation provides facilities for storing and retrieving various types of information.

In IMS 12, the IMS repository function provides facilities for storing and retrieving IMS resource definitions that are used in an IMSplex. The IMSplex can contain multiple IMS systems or a single IMS system. The IMS repository function will manage IMS resource definitions for both of these types of environments.

The ‘IMS repository function’ provides the architecture for a common method for storing and retrieving IMS definitions for both test and production environments. This IMS 12 implementation is the first usage of this infrastructure.

IMS Repository Function Usage

- The various components of the IMS repository function provide a centralized storage and retrieval solution for resource definitions
- 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, the 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

In IMS 12, the types of IMS resource definitions that can be managed by the IMS repository function are those definitions used by the DRD (Dynamic Resource Definition) function that provides for dynamic creation/maintenance of resource definitions for programs, transaction, databases and FP routing codes. This function originally became available in IMS 10.

This specific implementation of the repository function is called the IMSRSC, the IMS resource definition repository.

Previous to IMS 12, DRD implementation required the usage of RDDSs (resource definition data sets). Each IMS system has to have its own set of RDDSs, and coordination of these data sets in a multiple-IMS IMSplex environment had to be done manually. The IMSRSC repository in IMS 12 now provides another alternative for storing DRD definitions, the IMSRSC repository. This is a single, common repository that can be accessed by all IMSs in a multiple-IMS IMSplex configuration; manual coordination is no longer required since this ‘coordination’ is now provided by the IMS repository function itself.

An IMS user can now have their ‘stored resource definitions’ for DRD in either RDDSs or the repository. IMS will then use these stored definitions to generate ‘runtime resource definitions’ for programs, transactions, databases, and FP routing codes.

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
  - RDDS to / from repository utilities
  - DRD utilities

There are four major components of the IMS repository function in IMS 12.

1) A new Repository Server address space referred to as the ‘RS’ address space.
2) Repositories on DASD for storing definitions.
3) A CSL (Common Service Layer) IMSplex configuration.
4) Batch utilities to perform maintenance and migration/fallback functions.

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 or dynamically enable an IMSRSC repository via type-2 command
  - Can use the new IMSRSC repository to store definitions in IMS 12
- 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
  - Once migration is complete, use online commands to disable RDDSs

84

Current DRD users, either in IMS 10 or IMS 11 have three choices: 1) They can continue to use RDDSs with IMS 12, 2) They can migrate to IMS 12 with RDDSs, then migrate to the repository for DRD stored resource definitions (this is the preferred method), or 3) They can migrate to IMS 12 using the new repository functions for DRD stored resource definitions.

Users without DRD can 1) Implement DRD with RDDSs, 2) Implement DRD using the repository, or 3) Continue to use SYSGEN/online change to manage modblks definitions.

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
  - Repository can be implemented without an outage
- 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
  - Via commands (3 types)
  - Via batch utilities (3 functions)
- A strategic direction for IMS architecture

85

The main benefit of the IMS repository is that it consolidates IMS resource definitions in a single place that is IMS system-managed.

Environments using multiple-IMS IMSplexes gain significant benefits in the plex-wide coordination provided by the IMS repository. Single-IMS IMSplex environments gain benefits from having a comprehensive technique that provides improved management capabilities.

IMS Repository Function Infrastructure

This section covers more details about the major components of the IMS repository function.

IMS Repository Function components …

- A Repository Server (RS)
  - A 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
  - User must start the repository server address spaces

The first new component for the Repository function is a new address space called the repository server (RS) address space. It is BPE-based. Access to this RS address space is managed through the Resource Manager (RM) CSL manager.

There are two types of repository server address spaces: master and subordinate.

There is a single instance of a master repository server address in an IMSplex; it manages access to all repository data. It becomes the ‘master’ by being the first one started. Additional repository address spaces that are started become ‘subordinate’ repository server address spaces. They are used for recovery if the master repository server address space goes down. It is recommended to have one or more subordinate repository server address spaces.

For the rest of this presentation, the term ‘repository server’ address space will be used to refer to the master repository server.

IMS Repository Function components …

- A Repository Server (RS)
  - Uses VSAM KSDS data sets to store information
  - z/OS XCF is used 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
  - Recommend one master repository server address space per IMSplex

88

Data for the repository server is stored in VSAM KSDS data sets.

Communications with the repository server address space are via the z/OS XCF facility (not via the CSL SCI manager). Because XCF is used, the repository server address space can actually reside on any LPAR within the z/OS parallel sysplex where IMS systems are running though more typically it would reside on an LPAR in the IMSplex.

There can be only one master repository server per IMSplex. This master repository server can manage definitions for the IMSplex it resides in as well as definitions from other IMSplexes in the z/OS parallel sysplex environment. It is recommended that there be one master repository server address space per IMSplex for ease of management, though other configurations are possible.

IMS Repository Function components …

- A Repository Server (RS)
  - Has its own internal repository called the 'catalog repository'
  - Manages IMS repositories (IMSRSC for DRD)
  - Manages registrations and connections to the repository
  - 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

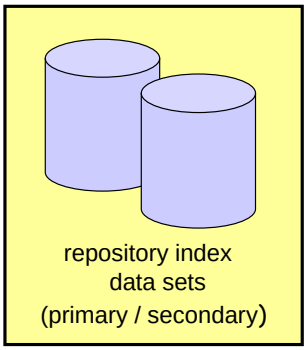
A repository server (RS) address space can manage multiple repositories. Each RS address space must have its own internal repository called the catalog repository. It can then manage multiple IMS repositories, such as the IMSRSC repository for DRD.

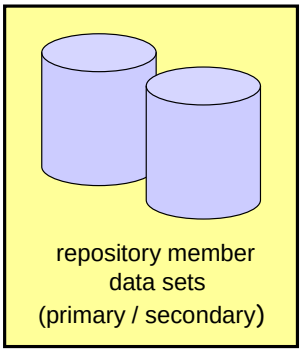
A repository server manages registrations and connections, provides data integrity for the information it manages, and uses SAF (RACF) interfaces for security.

Additionally, it proves an audit trail function of repository server activity using the z/OS logger. It also supports BPE tracing capabilities.

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
    - Catalog repository data sets
    - IMS repository data sets





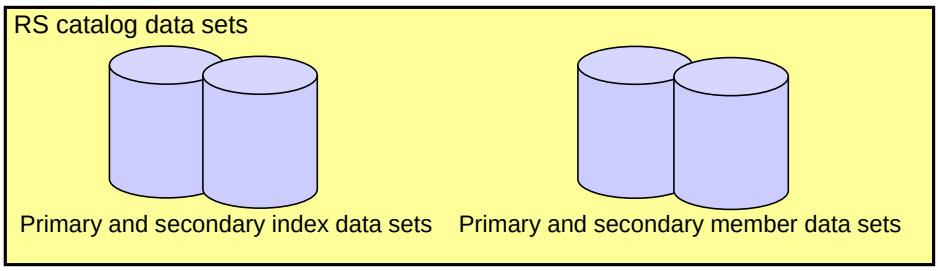
90

The second major component of the Repository function is the repositories.

Multiple sets of VSAM KSDS data sets are used to store the repository data. Each set has a repository index data set and a repository member data set. Each of these then has a primary and secondary data set plus in some cases a spare data set. Therefore each set of repository data sets is composed of at least 4 VSAM KSDS 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



91

The catalog repository data sets (RS catalog data sets) are used by the repository server itself to store its own data. These are required per repository server and must be allocated before a repository server can start up.

The RS catalog data sets always consist of 4 VSAM KSDS data sets, a primary and a secondary index data set, and a primary and secondary member data set. There is no spare capability for RS catalog data sets. These data sets are typically very small. If a write error occurs, the repository server itself will regenerate a new data set.

IMS Repository Function components …

- IMS repositories
  - 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)
        - Supports different attribute values for a particular resource definition in different IMSs (generic / IMS-specific)
    - Can have one or more IMSRSC repositories per Repository Server (RS)
      - Test
      - Production
    - Typically one IMSRSC repository per Repository Server (RS) per IMSplex

92

The IMS repository data sets store IMS resource definitions. In IMS 12, there is one type of IMS repository, the IMSRSC repository for managing DRD definitions.

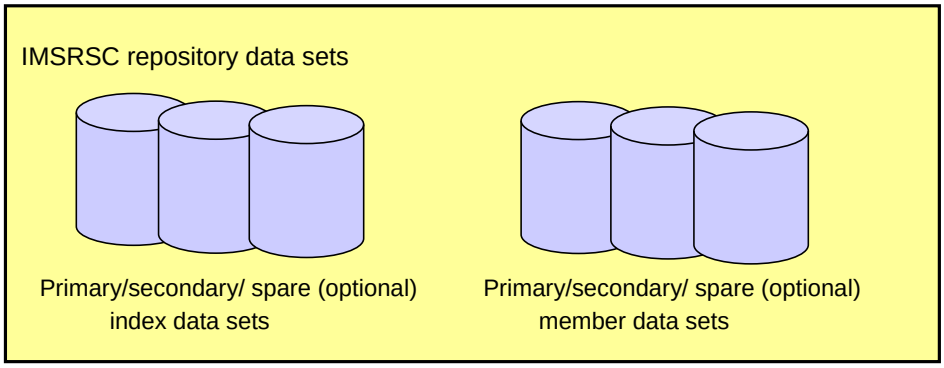
An IMSRSC repository contains two major types of information:

1) Resource lists for each IMS system that is using the repository server. These contain a list of all the programs, transactions, databases, and FP routing codes defined in a particular IMS system.

2) The actual stored resource definitions for all programs, transactions, databases, and FP routing code definitions that will be managed by the repository server. Different attribute values for a particular resource definition are supported by a generic definition and IMS-specific definitions where the attribute is different. An example of this is for the MSC SIDR and SIDL settings. Note that with RDDSs, manual processes are required to maintain different attribute values.

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) (COPY1)
      - Secondary index data set and secondary member data set (required) (COPY2)
      - Spare index data set and spare member data set (optional) (SPARE)



93

The IMSRSC repository is composed of either 4 or 6 VSAM KSDS data sets.

There will always be a primary index data set and a primary member data set, also referred to as COPY1 for some commands and messages.

There will always be a secondary index data set and a secondary member data set, also referred to as COPY2 for some commands and messages. .

Optionally, though recommended, there can be a spare index data set and a spare member data set, referred to as SPARE for some commands and messages. Spare data sets must be empty and are used for write failures on the primary/secondary copies.

IMS Repository Function components …

- IMS repositories
  - IMSRSC repository
    - States for repository data sets, maintained in RS catalog data sets
      - Primary (COPY1)
      - Secondary (COPY2)
      - Spare (SPARE)
      - Discarded (DISCARD)
    - IMSRSC repository must have valid primary/secondary sets of data sets or it is stopped
      - Recommend having spare pair of data sets

94

A particular IMSRSC repository index or member data set has a ‘state’, either primary (COPY1), secondary (COPY2), spare (SPARE), or discarded (DISCARD).

An IMSRSC repository must have valid primary/secondary sets of data sets at all times; if not, the IMSRSC repository is stopped. If a spare pair exists (recommended), then recovery will be performed.

IMS Repository Function components …

- IMS repositories
  - IMSRSC repository
    - 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’

A spare data set pair will be used when a write error is encountered either on the primary or secondary data set. The valid data set pair will be copied to the spare pair. The failed pair is marked ‘discarded’. During this recovery the repository is stopped, then restarted with the new primary/secondary copies. After the repository server successfully starts with the new primary/secondary copies, the user must define a new pair of empty spare data sets, and then use the batch ADMIN utility or repository server commands to set their status to ‘spare’, so they will be available if another write failure occurs.

Read errors are handled by using the other valid copy (primary or secondary) to access the needed data. There is no spare switching. This is an improvement for reads with the repository; RDDS and MODBLKS do not have a second data set.

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 via XCF
    - 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

96

The third major component of the Repository Server is a Common Service Layer (CSL) environment.

The Repository requires that a Common Service Layer (CSL) environment be available. The CSL needs to have an OM (Operations Manager) address space that will be used to process new and modified type-2 commands for repository functions. The CSL also needs to have an RM (Resource Manager) address space, which is used for access to the repository server (using z/OS XCF).

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 (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

97

The CSL also requires an SCI (Structured Call Interface) address space that is used for CSL communications. It is not used for communications between RM and the RS; these communications are done using z/OS XCF services.

The repository server can optionally register with SCI so it will show up on QUERY IMSPLEX output; it will register with SCI but that is the only SCI usage.

If available, a resource structure in the coupling facility will be used to provide repository name and repository type consistency within the IMSplex.

A SPOC is also required for entering certain types of repository commands.

This CSL needs to be available for both single-IMS IMplexes or a multiple-IMS IMSplexes.

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
  - DRD utilities (PM41218)
    - MODBLKS to IMSRSC repository
    - IMS log to IMSRSC repository

98

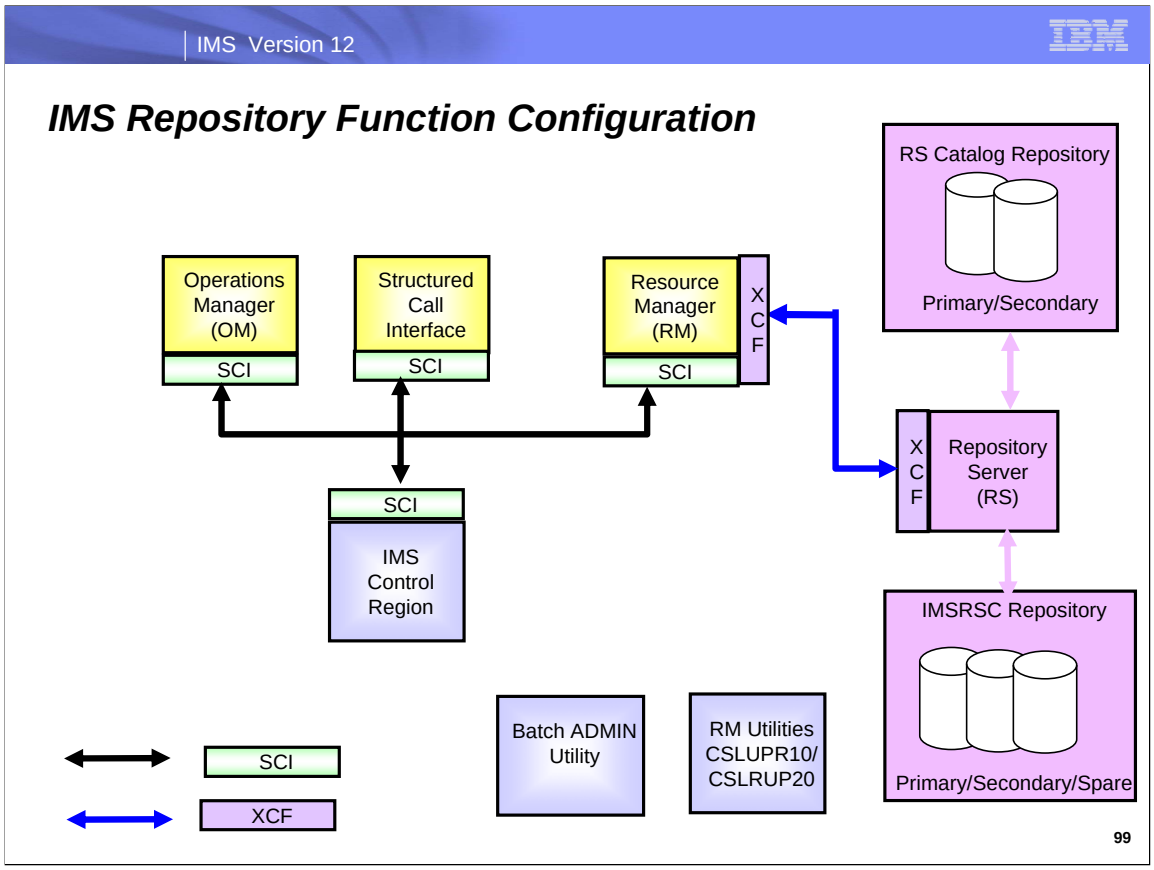
The last major component of the Repository function is the batch utilities that are provided.

A batch ADMIN utility (FRPBATCH) is provided to help manage the IMSRSC repositories that are part of a Repository Server. Some usage examples are to ADD a new IMSRSC repository to a repository server, to LIST the characteristics of an IMSRSC repository, and to START or STOP an IMSRSC repository.

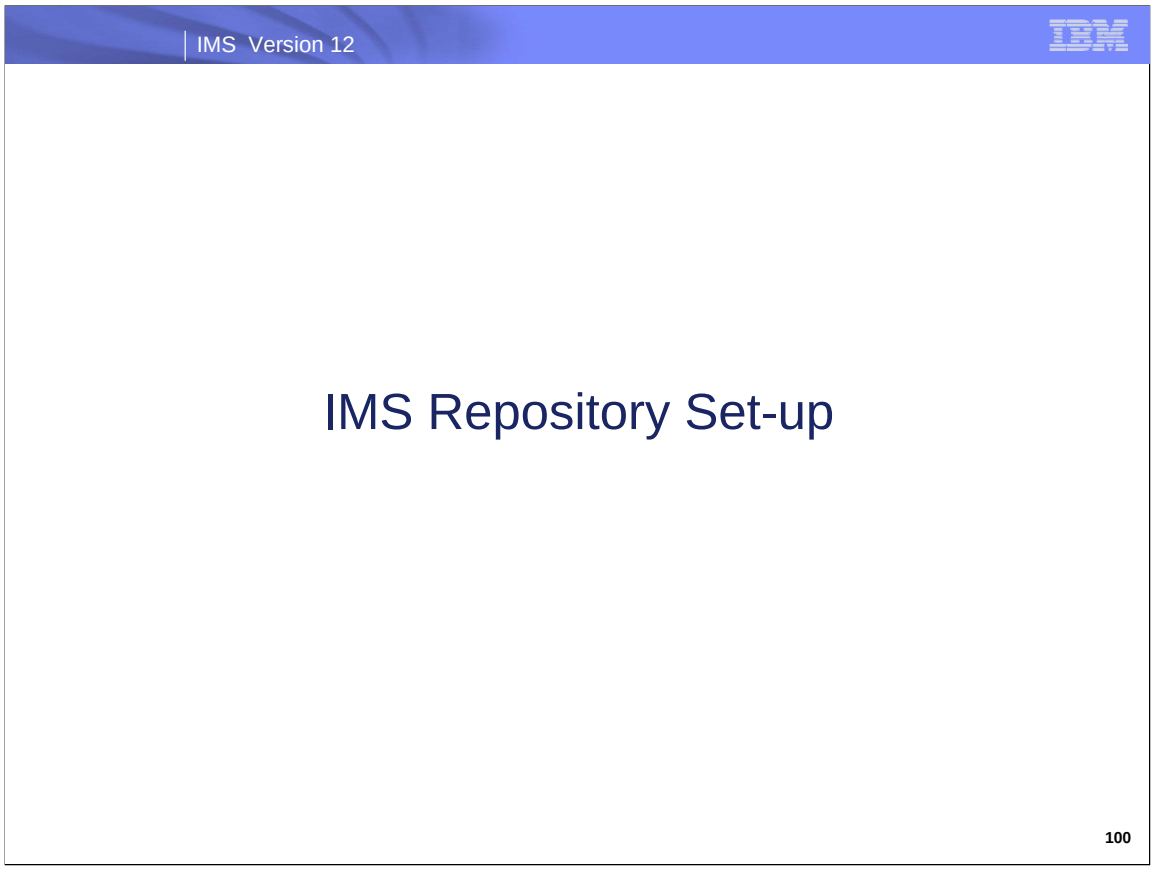
Two batch utilities (provided by the RM component) assist in migration to the repository server from RDDSs and fallback from the repository server to RDDSs.

CSLURP10 provides facilities to move definitions from an RDDS to the repository. CSLUPR20 provides facilities to move definitions from the repository to an RDDS.

The DRD utilities are enhanced via APAR PM41218 to provide MODBLKS to IMSRSC repository support and IMS log to IMSRSC repository support.



This diagram shows the major Repository Function components.



This section describes how to set-up the various components of the Repository in IMS 12.

IMS Repository Setup

- Repository Server
  - Create catalog repository data sets
  - Set up BPE configuration member
  - Set up FRPCFG configuration member
    - Repository server settings
    - Audit log definitions
    - Define security
  - Set up the CSL
  - Set up IMS
  - Start the master repository server
  - Start subordinate repository servers
- Enable an IMSRSC repository for DRD

101

There are several steps necessary to set up a repository server. This topic will cover the activities that need to be completed for each step. This process is only performed once for each repository server.

Once a repository server is set up, then you can enable an IMSRSC repository for DRD usage.

Create Catalog Repository Data Sets

- Catalog repository consists of two pairs of KSDSs
- Each pair has an ‘index data set’ and a ‘member data set’
  - Primary index data set and primary member data set
  - Secondary index data set and secondary member data set (duplex copy)
  - No support for spare (third) set of data sets
- Must be created before bringing up repository server (RS)
- The properties of the ‘secondary’ data sets must be the same as the ‘primary’ data sets
  - Recommend that the secondary be larger than the primary
  - Recommend that the primary and secondary be on different volumes/separate catalogs/storage controllers to ensure availability
- Provides the link between the repository name and VSAM data sets
- Sample in ‘Appendix’ for Repository Server setup/access

The catalog repository data sets must be created before a repository server can be started.

Considerations:

- The REUSE attribute is required, because it allows the Repository Server (RS) to automatically recover corrupt repository data sets at repository open time.
- You can use only share options (1,3) or (2,3). The share options for the INDEX and DATA components of a repository data set must match.
- The SPANNED attribute is not supported. Repository data sets must be non-spanned KSDSs.
- The recommended CI size is 8 KB for both the INDEX and DATA components of all of the repository data sets.
- The RS uses a single local shared resource (LSR) pool with a default 8 KB buffer size.
- For this buffer pool to be used optimally, make sure that the repository data sets have a matching 8 KB control interval size.

Recommendation**:** Define the primary, secondary and spare data set pairs on different volumes, to ensure availability. Ensure that the size of the secondary index and member data sets is greater than the size of the primary index and member data sets and that the size of the spare index and member data sets is greater than the size of the secondary index and member data sets

Setup BPE Configuration

- The Repository Server (RS) is a BPE-based address space
  - Must set up BPE first if not already available
    - Add BPEINI00 to the z/OS Program Properties Table (PPT)
- Two BPE PROCLIB members
  - Contain trace level and user exit information
  - BPE Configuration Parameters (BPECFG=BPExxxxx)
  - BPE User Exit List
    - EXITMBR= in BPECFG=
      - EXITDEF= in EXITMBR
  - Neither are required
    - Can let BPE configuration parameters default
    - Do not need user exits
  - All IMSplex members can share the BPE PROCLIB members

Since the repository server is a BPE-based address space, a BPE configuration must be available. This requires a z/OS PPT entry for BPE (most likely already there) and the set up for two BPE PROCLIB members. Using the defaults for these BPE PROCLIB members is recommended initially.

Setup BPE Configuration …

- Repository Server (RS) does support BPE tracing facilities
  - New BPE trace table type – DIAG
    - Contains diagnostic trace entries for the RS
  - REPO is the trace table owner

```
TRCLEV=(*,HIGH,REPO,PAGES=300) /* DEFAULT ALL TRACES TO HIGH */
```


The Repository Server does support BPE tracing facilities, including a new trace table type – DIAG.

Setup FRPCFG Configuration Member …

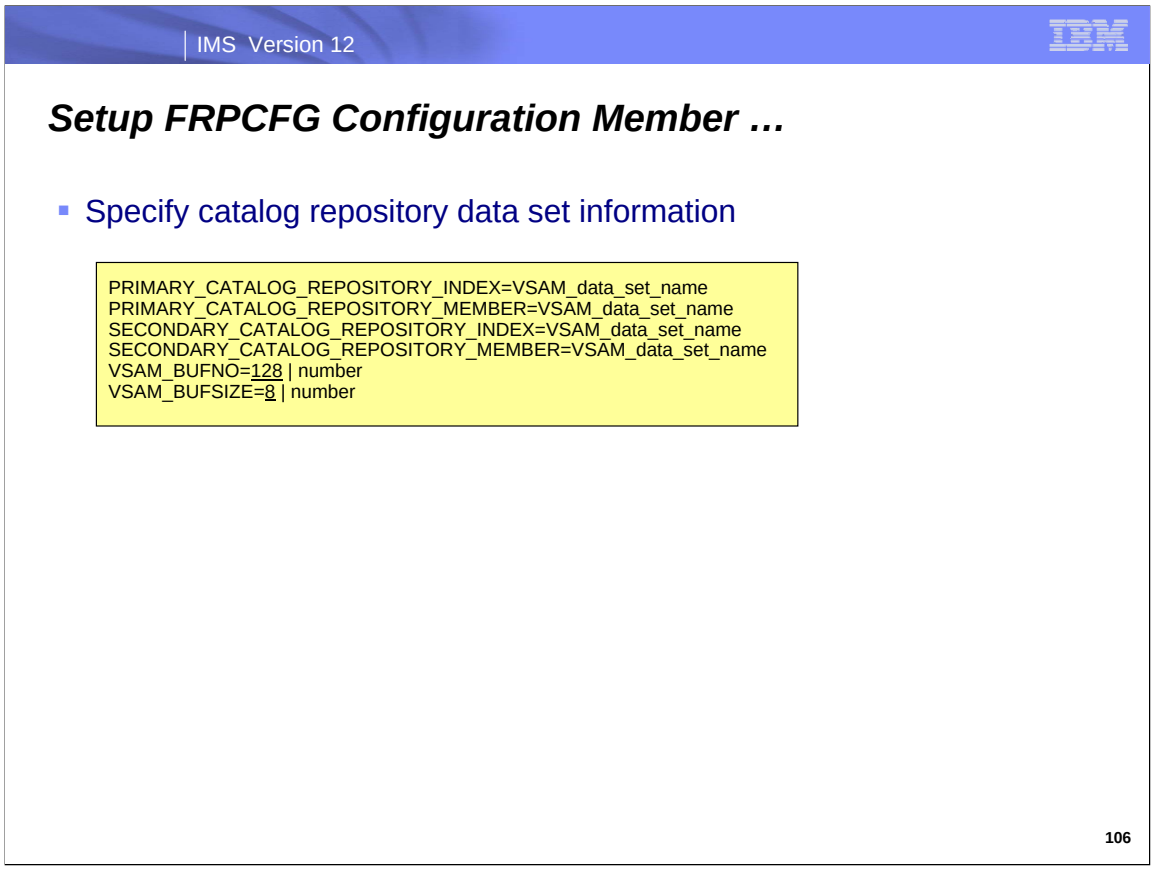
- Use FRPCFG configuration PROCLIB member to specify the repository server parameters
  - Catalog repository data set information
  - XCF settings
  - Configuration names (for SCI)
  - Security class
  - Auditing
- The PROCLIB member name to be used is specified by coding FRPCFG=*member_name* on the EXEC PARM= statement in the repository server address space startup JCL
- Sample repository server PROC called FRP

105

Repository server parameters are specified in the FRPCFG PROCLI B member for the repository server, via the FRPCFG= parameter in the RS address space startup JCL.

There are five types of information needed for a repository server

Catalog repository data set information
XCF settings
Configuration names (for SCI)
Security class
Auditing



For the catalog repository data sets, you must specify the names for the primary catalog repository index data set, the primary catalog repository member data set, and the names of their secondary copies. The repository server uses VSAM LSR for buffering using an 8K buffer size.

PRIMARY_CATALOG_REPOSITORY_INDEX=*VSAM_data_set_name*

This is a required keyword. Specifies the name of the primary repository index data set of the catalog. This name must match the name of the data set in the JCL that you use to create catalog repository data sets.

PRIMARY_CATALOG_REPOSITORY_MEMBER=*VSAM_data_set_name*

This is a required keyword. Specifies the name of the primary repository member data set of the catalog. This name must match the name of the data set in the JCL that you use to create catalog repository data sets.

SECONDARY_CATALOG_REPOSITORY_INDEX=*VSAM_data_set_name*

This is a required keyword. Specifies the name of the secondary repository index data set of the catalog. This name must match the name of the data set in the JCL that you use to create catalog repository data sets.

SECONDARY_CATALOG_REPOSITORY_MEMBER=*VSAM_data_set_name*

This is a required keyword. Specifies the name of the secondary repository member data set of the catalog. This name must match the name of the data set in the JCL that you use to create catalog repository data sets.

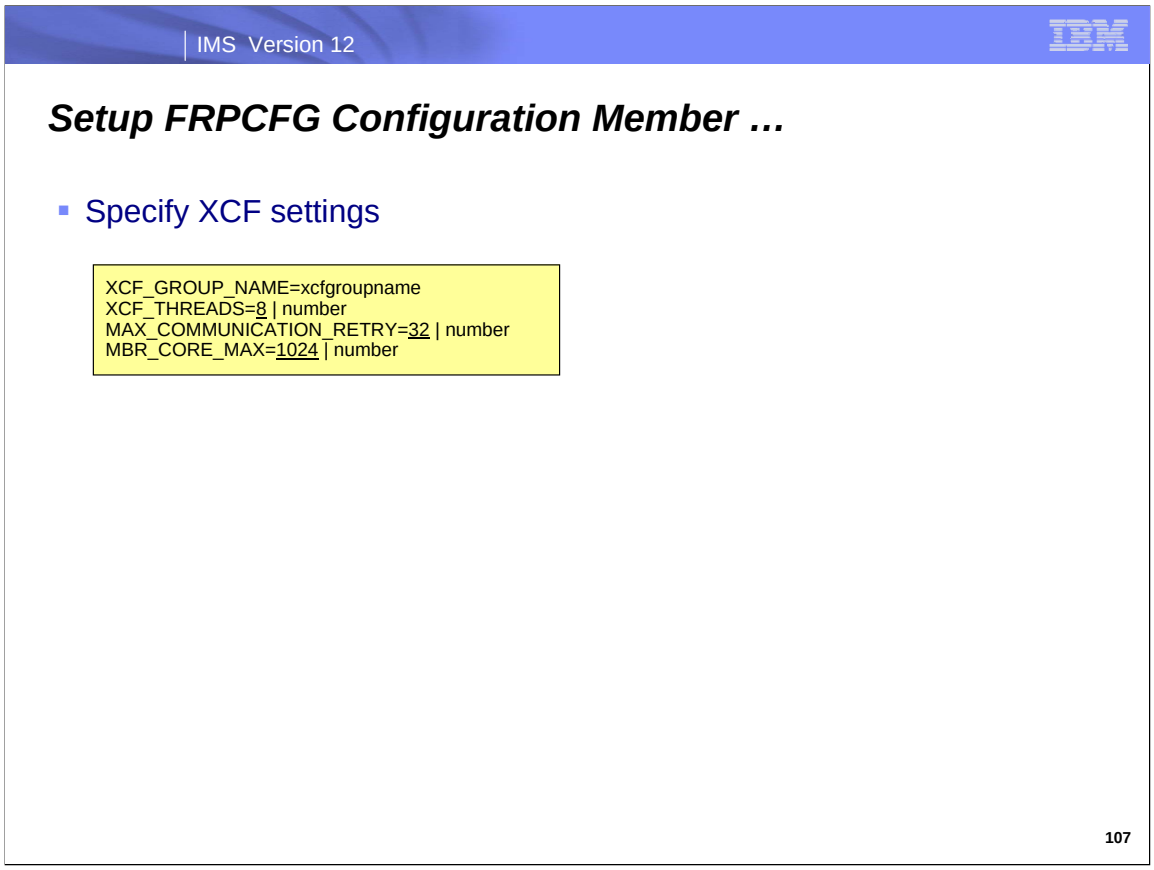
VSAM_BUFNO=128 | *number*

This keyword is optional. Specifies the number of VSAM buffers in the local shared resource pool used for repository access. All I/O between the RS and the repository index and member data sets uses a single local shared resource pool.. Valid values range from 3 to 65535. By default, 128 VSAM buffers are used.

VSAM_BUFSIZE=8 | *number*

This keyword is optional. Specifies the size, in KB, of the VSAM LSR pool buffers used for repository I/O. Enter a value from 8 to 32 that is a multiple of 4. By default, 8 KB buffers are used. In most cases, 8 KB is the optimal setting for both VSAM_BUFSIZE and the CONTROLINTERVALSIZE parameter in the RS startup JCL.

.



The XCF groupname must be specified; this groupname will be used for communications between the RS address space and the RM address space(s).

XCF_GROUP_NAME=*xcfgroupname*

This is a required keyword. The name of the XCF group that the server joins. Valid characters are A-Z (uppercase only), 0-9, and the following symbols: number sign (#), dollar sign ($), and at sign (@). The value must be eight characters padded on the right with blanks. For example, if you provide a value with only 4 characters, you must also include 4 blank spaces to the right of it.

XCF_THREADS=8 | *number*

This keyword is optional. Specifies the number of XCF listener threads available to accept data from clients. Each thread allocates a 32 KB buffer. Valid values range from 4 to 99. By default, 8 XCF threads are used.

MAX_COMMUNICATION_RETRY=32 | *number*

This keyword is optional. Specifies the number of times a client-side API process retries a failed communication when the failure is due to insufficient XCF threads. If this limit is exceeded, the client request fails with a reason code indicating that the server is busy. By default, 32 communication retries are used. Valid values range from 1 to 255.

MBR_CORE_MAX=1024 | *number*

This keyword is optional. Specifies the maximum amount, in KB, of incore storage allocated to support an XCF data package. If exceeded, a data space is created. By default, 1024 KB of incore storage is used. Valid values range from 64 to 4096.

Setup FRPCFG Configuration Member …

- Specify optional configuration names (for SCI and QUERY IMSPLEX)

```
IMSPLEX(NAME=imsplexname)
RSNAME=repositoryservername
```


The IMSPLEX and RSNAME parameters are used to indicate that the RS should register with SCI, so that the repository server will be included as an address space in the output of a QUERY IMSPLEX command.

IMSPLEX()

This keyword is optional. Specifies definitions for the IMSplex for the RS. Only one IMSPLEX keyword can be specified.

NAME=*name*

A 1-5 character identifier that specifies the IMSplex group name. This name defines the IMSplex that the RS will belong to. NAME= is required and no default exists. The RS concatenates this identifier to CSL to create the IMSplex group name.

When IMSPLEX() is specified, and when RSNAME= is also specified, the RS address space registers to the local SCI using the REPOID created from the RSNAME parameter as the SCI member name. Specifying IMSPLEX() and RSNAME= allows the RS to be shown on the output of the QUERY IMSPLEX command. **Note:** Including the RS in the QUERY IMSPLEX output is the only use of SCI made by the RS.

RSNAME=*repositoryservername*

This keyword is optional. Specifies the 1-6 character name for the RS address space. Specify this parameter either as an execution parameter or in the FRPCFG member of the IMS PROCLIB data set.

This name is used to create the REPOID, which is used in RS processing. The 8-character REPOID is the value for RSNAME followed by the characters RP. Trailing blank spaces in the value for RSNAME are deleted and the repository ID is padded on the right with blank spaces. For example, if RSNAME=ABC then the repository ID is ABCRP with 3 blank spaces on the right.

If RSNAME is specified, then the REPOID is appended to the end of all the FRPxx messages issued by the RS. If IMSPLEX() is also specified, then the REPOID is the member name used to register with SCI.

The RSNAME= in FRPCFG member should be unique for each Repository Server if there are multiple Repository Servers as SCI requires a unique member name

Setup FRPCFG Configuration Member …

- Specify optional security class name

```
SAF_CLASS=saf_class
```


SAF_CLASS=*saf_class*

This is an optional keyword. Specifies the 1- to 8-character SAF security class name, which is used to implement IMS repository and member-level security checking.

If this parameter is specified, *saf_class* must be the name of a defined resource class. If this parameter is omitted, SAF security is not used to restrict access to the RS.

The value must be a left-aligned 8-byte name with trailing contiguous spaces. The first character must be alphabetic and subsequent characters alphanumeric.

Setup FRPCFG Configuration Member

- Specify audit (audit log) settings ….

AUDIT=NO | YES

AUDIT_ID=id
AUDIT_LOG= z/OSlogstreamname

AUDIT_LEVEL=HIGH | NONE

AUDIT_DEFAULT=NOAUDIT | SECURITY | UPDATE | READ | SYSTEMREAD
AUDIT_FAIL=CONTINUE | ABORT

The audit parameters determine if the audit log exists and what information it contains.

AUDIT=N | Y This keyword is optional. Specifies whether auditing is enabled.

AUDIT=NO Auditing is not enabled. This is the default.

AUDIT=YES Enables auditing. If AUDIT=YES, AUDIT_ID and AUDIT_LOG are required.

AUDIT_ID=*id*

This keyword is required if AUDIT=YES. *id* is a unique number from 160 to 255 that is placed in the prefix of the log record to identify an audit source.

AUDIT_LOG= *z/OSlogstreamname* This keyword is required if AUDIT=YES. The name of the z/OS log stream to use for the audit records.

AUDIT_LEVEL=HIGH | NONE

This keyword is optional. Determines whether audit records are written to the log. You can change the audit level dynamically by using the command F reposervername,AUDIT.

HIGH Audit records are written to the log. This is the default.

NONE Audit records are not written to the log.

Setup FRPCFG Configuration Member

- Specify audit (audit log) settings

AUDIT=NO | YES

AUDIT_ID=id
AUDIT_LOG= z/OSlogstreamname

AUDIT_LEVEL=HIGH | NONE

AUDIT_DEFAULT=NOAUDIT | SECURITY | UPDATE | READ | SYSTEMREAD
AUDIT_FAIL=CONTINUE | ABORT

The audit parameters determine if the audit log exists and what information it contains. The audit levels listed in the AUDIT_DEFAULT keyword create an increasing number of audit log entries. SYSTEMREAD will create the most and is the highest level of auditing.

AUDIT_DEFAULT=NOAUDIT | SECURITY | UPDATE | READ | SYSTEMREAD

This is an optional keyword. Determines the default level of auditing of member access during a client member session. This access level applies to members of a given repository for which no audit access rule is set.

NOAUDIT No auditing of member access. This the default.

SECURITY Audit security failures only.

UPDATE Audit member access with update intent.

READ Audit member access with read or update intent. Under an audit access rule of READ, system read requests do not cause a read audit record to be generated.

SYSTEMREAD Audit member access with system-level read, read, or update intent. A read of the resource definition by the system before the update request is identified as a 'system read' request. Under an audit access rule of SYSTEMREAD, all read requests, including system read requests, are audited.

The value for the AUDIT_DEFAULT parameter can be overridden by setting the AUDITACCESS parameter in the CSLRIxxx member of the IMS PROCLIB data set.

AUDIT_FAIL=CONTINUE | ABORT

This keyword is optional. Determines whether the RS continues to start or aborts if auditing cannot be established because the log stream is unavailable. After the RS has started, this setting determines what happens if a client audit request is unsuccessful.

CONTINUE The client request continues as if the audit request was successful. This is the default.

ABORT The client request fails if the audit record cannot be written.

Repository Server Audit Log Setup

- Auditing specified via AUDIT=YES parameter in FRPCFG configuration PROCLIB member in repository server JCL
- Can turn off/on auditing via repository server AUDIT command
- Examples of events that create audit log entries
  - Security errors
  - Registration and connection events
  - Access to repository
- Mapping macro FRPLGREC provides audit log record formats
- Can print records from the repository server audit log with DFSERA10
- No facility for deleting records from the repository server audit trail
- No facility for viewing records in the repository server audit trail
- Recommend using a separate z/OS log stream for the repository server

112

To have an audit log, you specify the AUDIT=YES parameter in the FRPCFG configuration PROCLIB member for the repository server. Auditing can be turned off or back on with the repository server AUDIT command (via z/OS modify interface).

The repository server audit log can be printed using DFSERA10 with exit CSLRERA3. There are no delete or viewing capabilities provided by IMS.

Repository Server Audit Log Setup

- z/OS logger setup
  - LOGR policy

```
DEFINE STRUCTURE NAME(MVSLOGREPO01)
       LOGSNUM(1)
       AVGBUFSIZE(4096)
       MAXBUFSIZE(65272)

DEFINE LOGSTREAM NAME(SYSLOG.REPO01.LOG)
       STRUCTNAME(MVSLOGREPO01)
       HLQ(IXGLOGR)
       LS_STORCLAS(LOGGER1)
       LS_DATACLAS(LOGGER1)
       LS_MGMTCLAS(LOGGER)
       LS_SIZE(50)
```

  - CFRM policy

```
STRUCTURE NAME(MVSLOGREPO01)
         SIZE(12288)
         INITSIZE(8192)
         PREFLIST(LF03)
```


Here is an example of the z/OS logger setup required for using the repository server audit log.

Printing the Repository Server Audit Log

- To print the Repository Server (RS) audit log, use DFSERA10 with exit CSLRERA3, the RM Repository Audit Log Print Exit

```
//CSLERA1 JOB MSGLEVEL=1,MSGCLASS=A,CLASS=K
//STEP1 EXEC PGM=DFSERA10
//STEPLIB DD DISP=SHR,DSN=IMS.SDFSRESL
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD DSN=SYSLOG.REPO.AUDIT.LOG,
// SUBSYS=(LOGR,IXGSEXIT),
// DCB=(BLKSIZE=32760)
//SYSIN DD
CONTROL CNTL H=EOF
OPTION PRINT EXITR=CSLRERA3
END
//
```

- Sample in ‘Appendix’ for Repository Server setup/access

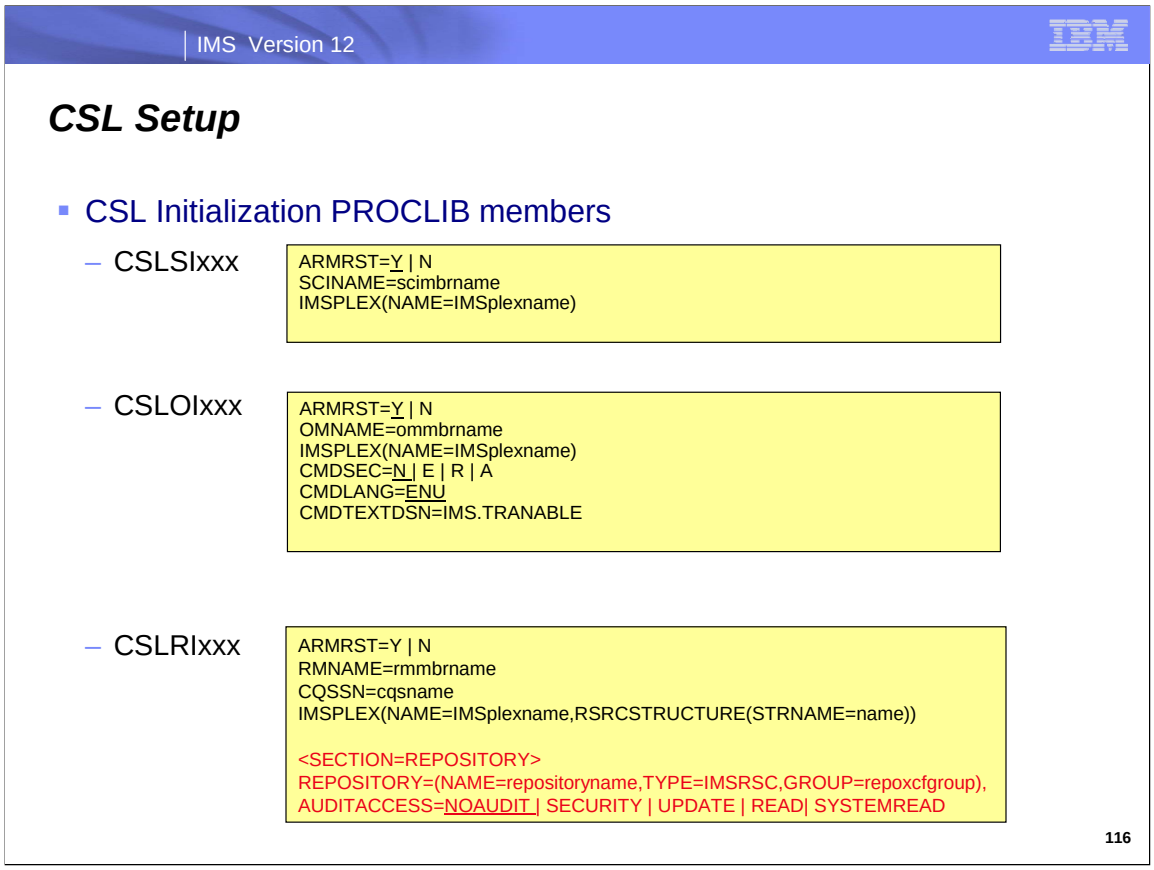
This is an example of the JCL that would be used to print records from the repository server audit log and an example of the output.

Setup Repository Server Configuration

- Sample FRPCFG configuration PROCLIB member

```
PRIMARY_CATALOG_REPOSITORY_INDEX=IMSPLXA.CAT.REPO1.RID
PRIMARY_CATALOG_REPOSITORY_MEMBER=IMSPLXA.CAT.REPO1.RMD
SECONDARY_CATALOG_REPOSITORY_INDEX=IMSPLXA.CAT.REPO2.RID
SECONDARY_CATALOG_REPOSITORY_MEMBER=IMSPLXA.CAT.REPO2.RMD
VSAM_BUFNO=128
VSAM_BUFSIZE=8
XCF_GROUP_NAME=FRPSPLEX
XCF_THREADS=4
MBR_CORE_MAX=1024
IMSPLEX(NAME=PLEX1)
RSNAME=REPO1
#SAF_CLASS=FACILITY
AUDIT=YES
AUDIT_ID=160
AUDIT_LOG=SYSLOG.REPO01.LOG
AUDIT_LEVEL=HIGH
AUDIT_DEFAULT=SYSTEMREAD
AUDIT_FAIL=CONTINUE
```


Here is an example of the FRPCFG configuration PROCLIB member for the repository server.



CSL initialization PROCLIB members must be set up when using the repository server. The SCI initialization member (CSLSIxxx) and the OM initialization member (CSLOMxxx) just need to contain standard CSL parameters (in black).

The RM initialization PROCLIB member (CSLRIxxx) contains the standard CSL parameters (in black) plus some unique parameters for using the repository server (in red).

There is a REPOSITORY section that must come after the other RM initialization parameters. This specifies the name of the IMSRSC repository, the XCF groupname, and optional audit log access parameters. There can only be one REPOSITORY= statement.

REPOSITORY=()

Defines the IMS repository parameters for RM initialization. It is specified within a section with the header <SECTION=REPOSITORY>.

NAME=

Specifies the repository name that is managed by RM. This name must be same as the repository name defined to the Repository Server (RS) in the ADD REPOSITORY function. The repository name can be up to 44 characters long and can contain the alphanumeric characters (A–Z, 0 - 9) and the following symbols: period (.) at sign (@), number sign (#), underscore (_), and dollar sign ($). **Note:** The alphabetic characters A–Z can be uppercase only.

A repository name of CATALOG cannot be specified, because it is reserved for use by the RS.

TYPE=

Specifies the repository type. The only valid value is IMSRSC.

GROUP=

Specifies the Repository Server z/OS cross-system coupling facility group name. This value must be the same as the XCF group name specified on the XCF_GROUP_NAME parameter of the FRPCFGxxx member. RM and the RS must be in the same XCF group. The value must be 8 characters padded on the right with blanks. Valid characters are A-Z (uppercase only), 0 - 9, and the following symbols: number sign (#), dollar sign ($), and at sign (@).

AUDITACCESS=

An optional parameter. It specifies the repository audit access level for the specified repository. If this value is not specified, the audit access level defaults to the level of auditing set by the AUDIT_DEFAULT= parameter in the FRPCFGxxx member of the IMS PROCLIB data set. The valid values for the AUDITACCESS= parameter are:

DEFAULT Use the rule specified in the AUDIT_DEFAULT parameter of the FRPCFGxxx member.
NOAUDIT No auditing of member access.
SECURITY Audit security failures only.
UPDATE Audit member access with update intent.
READ Audit member access with read or update intent.
SYSTEMREAD
Audit member access with system-level read, read, or update intent.

A read of the resource definition by the system before the update request is identified as a "system read" request.

Under an audit access rule of READ, system read requests do not cause a read audit record to be generated.

Under an audit access rule of SYSTEMREAD, all read requests, including system read requests, are audited.

CSL Setup …

- CSL PROCs
  - CSLSCI

```
//SCI1  PROC              RGN=0,SOUT=A, RESLIB='IMS.SDFSRESL',
//                          BPECFG=BPEPLX0,
//                          SCIINIT=001,
//                          PARM1=
//SCIPROC    EXEC         PGM=BPEINI00,REGION=&RGN,
//                          PARM='BPECFG=&BPECFG,
//                          BPEINIT=CSLSINI0,
//                          SCIINIT=&SCIINIT,&PARM1'
//STEPLIB                 DD    DSN=&RESLIB,DISP=SHR
//   ..
```

  - CSLOM

```
//OM1  PROC               RGN=0,SOUT=A, RESLIB='IMS.SDFSRESL',
//                          BPECFG=BPEPLX0,
//                          OMINIT=001,
//                          PARM1=
//OMPROC  EXEC            PGM=BPEINI00,REGION=&RGN,
//                          PARM='BPECFG=&BPECFG,
//                          BPEINIT=CSLOINI0,
//                          OMINIT=&OMINIT,&PARM1'
//STEPLIB                 DD    DSN=&RESLIB,DISP=SHR
//   ..
```

117

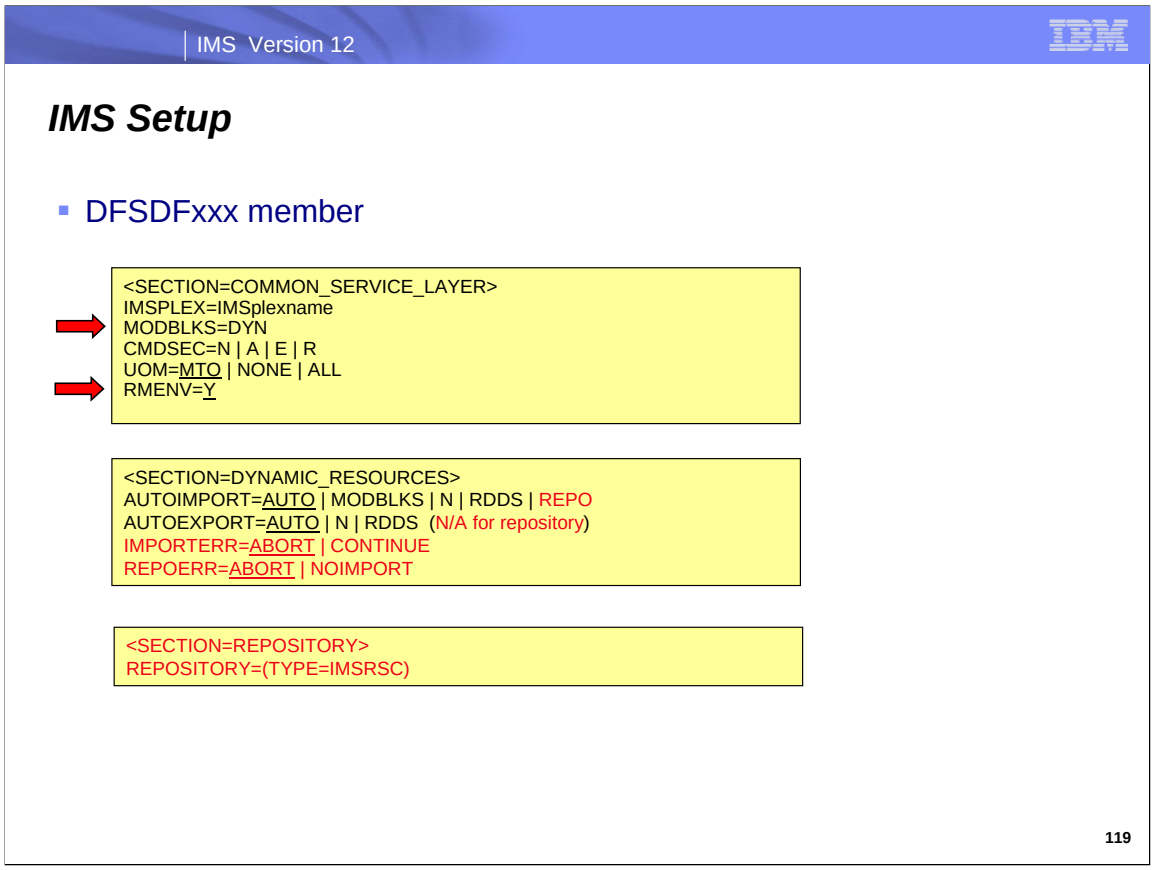
These are the standard CSL procedures for SCI and OM.

CSL Setup

- CSL PROCs
  - CSLRM

```
//RM1  PROC              RGN=0,SOUT=A, RESLIB='IMS.SDFSRESL',
//                         BPECFG=BPEPLX0,
//                         RMINIT=001,
//                         PARM1=
//RMPROC  EXEC           PGM=BPEINI00,REGION=&RGN,
//                         PARM='BPECFG=&BPECFG,
//                         BPEINIT=CSLRINI0,
//                         RMINIT=&RMINIT,&PARM1'
//STEPLIB                DD   DSN=&RESLIB,DISP=SHR
//   ..
```


This is the standard CSL procedure for RM.



The DFSDFxxx PROCLIB member for IMS systems has some special parameters (in red) for repository use.

In the Common Service Layer section,
- specify MODBLKS=DYN to indicate DRD usage (vs. OLC)
- specify RMENV=Y to indicate that an RM address space will be used.

In the Dynamic Resources section,
- AUTOIMPORT=REPO is a new option for autoimport
- AUTOEXPORT= does NOT apply to the repository – need to do EXPORT command
- IMPORTERR=ABORT | CONTINUE applies to the repository as well as RDDSs
- REPOERR=ABORT | NOIMPORT specifies how to handle errors when using the repository (vs. RDDSs).

There is a new Repository section that specifies the type of IMS repository that will be used.
- TYPE=IMSRSC is the only valid option. There can be only one of these statements.

AUTOIMPORT=AUTO | MODBLKS | NO | RDDS | REPO

Specifies whether resource and descriptor definitions are automatically imported during IMS cold start.

Automatic import from the IMS repository is enabled if all of the following conditions are true:
- IMS is enabled with DRD.
- The repository section of the DFSDFxxx member is defined with TYPE=IMSRSC.
- IMS is enabled with RM services (RMENV=N is not specified in the DFSCGxxx member of the IMS PROCLIB data set or in the CSL section of the DFSDFxxx member)
- The CSLRIxxx member is defined for the repository with TYPE=IMSRSC.
- The repository contains stored resource definitions for the IMS system.

REPOERR=ABORT | NOIMPORT

Specifies the action to perform if there are errors importing data from an IMS repository that are not due to invalid or missing resource or descriptor definitions. A DFS4401E message is issued with the Resource Manager (RM) request return and reason code.

If there are errors importing data from a repository that are due to invalid or missing resource or descriptor definitions, the IMPORTERR= parameter determines the action to perform.

ABORT Abort the IMS cold start if there are errors importing data from a repository. ABORT is the default.

NOIMPORT Continue the IMS cold start with no resources imported from a repository.

Start the master repository server

```
//*****************************************************************
//*     FRP Procedure                                             *
//*                                                               *
//*                                                               *
//*     Parameters:                                               *
//*     BPECFG  - Name of BPE member                              *
//*     FRPCFG  - Name of the FRPCONFG member                     *
//*                                                               *
//*************************************************************@SCPYRT**
//*                                                               *
//*       Licensed Materials - Property of IBM                    *
//*                                                               *
//*       "Restricted Materials of IBM"                           *
//*                                                               *
//*       5635-A03 (C) Copyright IBM Corp. 2010                   *
//*                                                               *
//*************************************************************@ECPYRT**
//*
//FRP      PROC RGN=3000K,SOUT=A,
//             RESLIB='IMS.SDFSRESL',
//             BPECFG=BPECONFG,
//             FRPCFG=FRPCONFG
//*
//FRPPROC  EXEC PGM=BPEINI00,REGION=&RGN,
//  PARM='BPEINIT=FRPINI00,BPECFG=&BPECFG,FRPCFG=&FRPCFG'
//*
//STEPLIB  DD  DSN=&RESLIB,DISP=SHR
//         DD  DSN=SYS1.CSSLIB,DISP=SHR
//*
//PROCLIB  DD  DSN=IMS.PROCLIB,DISP=SHR
//*
//FRPPRINT DD  SYSOUT=&SOUT
//SYSPRINT DD  SYSOUT=&SOUT
//SYSUDUMP DD  SYSOUT=&SOUT
//*
//*-----------------------------------------------------------------*
//* End of Procedure FRP  ..........                                *
//*-----------------------------------------------------------------*
```


Here is an example of thr FRP procedure that can be used to start the first repository server. A successful start receives the message FRP2002I Master repository server status obtained. It becomes the master repository server.

Start the subordinate repository server(s)

```
//*****************************************************************
//*     FRP Procedure                                             *
//*                                                               *
//*                                                               *
//*     Parameters:                                               *
//*     BPECFG  - Name of BPE member                              *
//*     FRPCFG  - Name of the FRPCONFG member                     *
//*                                                               *
//*************************************************************@SCPYRT**
//*                                                               *
//*       Licensed Materials - Property of IBM                    *
//*                                                               *
//*       "Restricted Materials of IBM"                           *
//*                                                               *
//*       5635-A03 (C) Copyright IBM Corp. 2010                   *
//*                                                               *
//*************************************************************@ECPYRT**
//*
//FRP      PROC RGN=3000K,SOUT=A,
//             RESLIB='IMS.SDFSRESL',
//             BPECFG=BPECONFG,
//             FRPCFG=FRPCONFG
//*
//FRPPROC  EXEC PGM=BPEINI00,REGION=&RGN,
//  PARM='BPEINIT=FRPINI00,BPECFG=&BPECFG,FRPCFG=&FRPCFG'
//*
//STEPLIB  DD  DSN=&RESLIB,DISP=SHR
//         DD  DSN=SYS1.CSSLIB,DISP=SHR
//*
//PROCLIB  DD  DSN=IMS.PROCLIB,DISP=SHR
//*
//FRPPRINT DD  SYSOUT=&SOUT
//SYSPRINT DD  SYSOUT=&SOUT
//SYSUDUMP DD  SYSOUT=&SOUT
//*
//*-------------------------------------------------------------------*
//* End of Procedure FRP  ..........                                  *
//*-------------------------------------------------------------------*
```

121

Here is an example of JCL that can be used to start subordinate repository servers. A successful start receives message FRP2001I Subordinate repository server status obtained. The subordinate repository server(s) must have the same XCF group name specified in their FRPCFG= PROCLIB member. It must have a different RSNAME parameter also.

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

After the repository server is setup and started, then you need to set up an IMSRSC repository that can be used for DRD definitions. This chart lists the steps that need to be done.

Creating IMSRSC repository data sets

- IMSRSC repository data sets
  - Primary index data set and primary member data set
  - Secondary index data set and secondary member data set
  - Spare index data set and spare member data set (optional but recommended)
    - Will be empty unless needed if write error occurs on primary/secondary
- Can have multiple IMSRSC repositories per repository server
  - Example, test and production
- The properties of the ‘secondary’ and ‘spare’ data sets must be the same as the ‘primary’ data sets
  - Recommend that the secondary be larger than the primary, the spare be larger than the secondary
  - Recommend that the primary, secondary, and spare be on different volumes/separate catalogs/storage controllers/etc. for availability
  - Plan for VSAM data set maintenance
- Sample in ‘Appendix’ for Repository Server setup/access

123

After the repository server is setup and started, the IMSRSC repository data sets need to be created. There can be multiple IMSRSC repositories per repository server; however, the more typical configuration will have one IMSRSC repository per repository server per IMSplex.

Considerations:

- The REUSE attribute is required, because it allows the Repository Server (RS) to automatically recover corrupt repository data sets at repository open time.
- You can use only share options (1,3) or (2,3). The share options for the INDEX and DATA components of a repository data set must match.
- The SPANNED attribute is not supported. Repository data sets must be non-spanned KSDSs.
- The recommended CI size is 8 KB for both the INDEX and DATA components of all of the repository data sets.
- The RS uses a single local shared resource (LSR) pool with a default 8 KB buffer size.
- For this buffer pool to be used optimally, make sure that the repository data sets have a matching 8 KB control interval size.

Recommendation**:** Define the primary, secondary and spare data set pairs on different volumes, to ensure availability. Ensure that the size of the secondary index and member data sets is greater than the size of the primary index and member data sets and that the size of the spare index and member data sets is greater than the size of the secondary index and member data sets

Define Security for IMS Repository

- 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

This chart summarizes the types of security facilities available for securing the repository environment. More security details will be discussed in Repository Part 2.

Define the IMSRSC Repository to the repository server

- Use the batch ADMIN utility ADD command to place the IMSRSC repository definition into the repository server catalog repository

```
ADD REPOSITORY(IMS_REPOS) +
REPDSN1RID(IMSPLXA.IMSRSC.REPO1.RID) +
REPDSN1RMD(IMSPLXA.IMSRSC.REPO1.RMD) +
REPDSN21RID(IMSPLXA.IMSRSC.REPO2.RID) +
REPDSN2RRMD(IMSPLXA.IMSRSC.REPO2.RMD) +
REPDSN3RID(IMSPLXA.IMSRSC.REPO3.RID)  +
REPDSN3RMD(IMSPLXA.IMSRSC.REPO3.RMD) +
AUTOOPEN(NO)
```


The definition for this IMSRSC repository needs to be placed into the repository server catalog repository using the batch ADMIN utility ADD command.

Start the IMSRSC repository

- Use the batch ADMIN utility START command to make the IMSRSC repository available for use

```
START REPOSITORY(IMS_REPOS) MAXWAIT(30,CONTINUE)
```


The IMSRSC repository can be made available for use by issuing the batch ADMIN utility START command.

Enable the IMSRSC repository to RM and IMS

- Issue the type-2 UPDATE RM command to dynamically enable the IMSRSC repository for DRD usage
  - CSLRIxxx contains repository section

```
UPDATE RM TYPE(REPO) REPOTYPE(IMSRSC) SET REPO(Y)
```

- Issue the type-2 UPDATE IMS command to enable IMS to use the repository
  - DFSDFxxx contains repository section

```
UPDATE IMS SET(LCLPARM(REPO(Y) REPOTYPE(IMSRSC)
```


The next step is to enable the IMSRSC repository.

You need to dynamically allow RM to start using the IMSRSC repository by issuing the type-2 UPDATE RM command. You need to enable IMS to use the IMSRSC repository by issuing the type-2 UPDATE IMS command.

Populate the IMSRSC repository

- If IMS is up, use an empty repository
  - Then populate it using the EXPORT command to add DRD stored resource definitions to the repository

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

- If IMS is down, use the batch RDDS to Repository Utility (CSLURP10) to populate the IMSRSC repository
  - Ensures no DRD changes are being made to the RDDS

```
//RDDS2RPO JOB ,USER,CLASS=A,MSGCLASS=X,NOTIFY=USER
//*
//STEP1 EXEC PGM=CSLURP10,MEMLIMIT=4G
//SYSUDUMP DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//RDDSDSN DD DSN=IMSTESTG.IMSRDDS1,DISP=SHR
//SYSIN DD *
IMSPLEX(NAME=PLEX1 IMSID(SYS3,IMS2,IMS3))
//*
```

  - Coldstart IMS with AUTOIMPORT=REPO or AUTO to import from repository
  - Coldstart IMS with AUTOIMPORT=NO or warm/emergency restart IMS, issue IMPORT command

```
IMPORT DEFN SOURCE (REPO) TYPE(ALL) NAME(*)
```

128

The next step is to populate the IMSRSC repository.

If you are doing this while IMS is up, the repository should be empty; then the EXPORT command can be used to write the DRD stored resource definitions to the repository.

If you are doing this while IMS is down, the RDDS to Repository Utility (CSLURP10) would be run to move definitions from an RDDS to the IMSRSC repository.

Then you can either AUTOIMPORT the DRD definitions from the repository if AUTOIMPORT is being used for a coldstart. If AUTOIMPORT is not being used on a coldstart or IMS is restarting with a warm/emergency restart, then you would issue the IMPORT command to read the DRD definitions from the repository.

IMSRSC repository is active and 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)
```


The IMSRSC repository is now available for use for DRD stored resource definitions for programs/transactions/databases/FP routing codes using type-2 commands.

Repository Startup Sequence Guidelines

- Recommended start up sequence with all CSL components
  - SCI
  - OM
  - CQS
  - RS
  - RM
  - IMS
    - DBRC
  - SPOC
  - ODBM
  - IMS Connect

130

The repository server address space must be up before RM comes up (RM initialization PROCLIB member CSLRIxxx with REPOSITORY section). Then IMS can come up (uses DFSDFxxx PROCLIB member sections). If IMS comes up before RM with RM not enabled for a repository, IMS will abend with a U400.

IMS Repository Server Access

This topic will discuss the communication methods that are used to access the repository server.

Accessing the IMS Repository Server

- Access through RM address space
  - Online
    - IMS type-2 commands
  - Batch RM utilities
    - RDDS to Repository (CSLURP10)
    - Repository to RDDS (CSLURP20)
- Direct access (RM not used)
  - Batch ADMIN utility (FRPBATCH)
  - Repository server commands using the z/OS modify interface

Most access to the IMS repository server will be through the RM address space, both online via type-2 commands and in batch via two RM utilities.

There are two facilities that do not use RM to access the repository. These are the batch ADMIN utility (FRPBATCH) that uses XCF and a set of repository server commands that use the z/OS modify interface. These facilities are used for repository management.

IMS Repository Access through RM address space

- Online operation
  - IMS type-2 commands
    - UPDATE RM
    - QUERY RM
    - UPDATE IMS
    - QUERY IMS

    For dynamic enablement / disablement of repository and RDDS functions

    For status of RM and 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
    - DRD commands (CREATE, UPDATE, DELETE) work with runtime definitions, not the stored resource definitions in the repository

133

During onlne operation, there are several new and modified IMS type-2 commands that access the repository through RM. UPDATE RM, QUERY RM, UPDATE IMS, QUERY IMS are used for dynamic enablement and disablement of repository and RDDS functions. EXPORT DEFN, IMPORT DEFN, DELETE DEFN are used for working with DRD stored resource definitions in the repository. The QUERY command can display the actual stored resource definition with its attributes. Remember that the DRD CREATE, UPDATE, and DELETE commands work with runtime definitions, not the stored resource definitions in the repository.

IMS Repository Access through RM address space

- Batch RM utilities
  - RDDS to Repository Utility (CSLURP10)
    - Utility to write the contents of a system or non-system RDDS to an IMSRSC repository
      - To initially populate an IMSRSC repository
      - To update definitions in an IMSRSC repository
    - Requires SCI and RM
  - Repository to RDDS Utility (CSLUPR20)
    - Utility to generate a non-system RDDS from an IMSRSC repository
      - Used for backup and migration/fallback
    - Requires SCI and RM
- Samples in 'Appendix' for Repository Server setup/access

134

Two batch RM utilities, CSLURP10 and CSLURP20, are provided for migration/fallback with the repository. They access the repository through RM.

CSLURP10 takes the contents of a system or non-system RDDS and writes these definitions to an IMSRSC repository. This utility can be used to initially populate an IMSRSC repository as well as update definitions at some later time. Since CSLURP10 uses RM to communicate with the repository, an SCI address space and an RM address space must be available.

CSLURP20 generates a non-system RDDS from an IMSRSC repository. It can be used for backup and migration/fallback. Since CSLURP20 uses RM to communicate with the repository, an SCI address space and an RM address space must be available.

IMS Repository Direct Access (RM not used)

- Batch ADMIN utility – 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
- Sample in 'Appendix' for Repository Server setup/access

135

A repository batch ADMIN utility (FRPBATCH) is provided to help manage IMSRSC repositories. There are 8 commands provided. This utility does not use RM to access the repository; rather it accesses the repository directly as a batch job using XCF. This information about IMSRSC repositories is kept in the catalog repository.

The ADD, UPDATE, RENAME, and DELETE commands provide the capability to manage IMSRSC repository definitions. The DSCHANGE command provides the capability to change a data set disposition (used to set up new SPARE data sets). The LIST command provides a display of IMSRSC information. START and STOP allow/stop access to an IMSRSC repository.

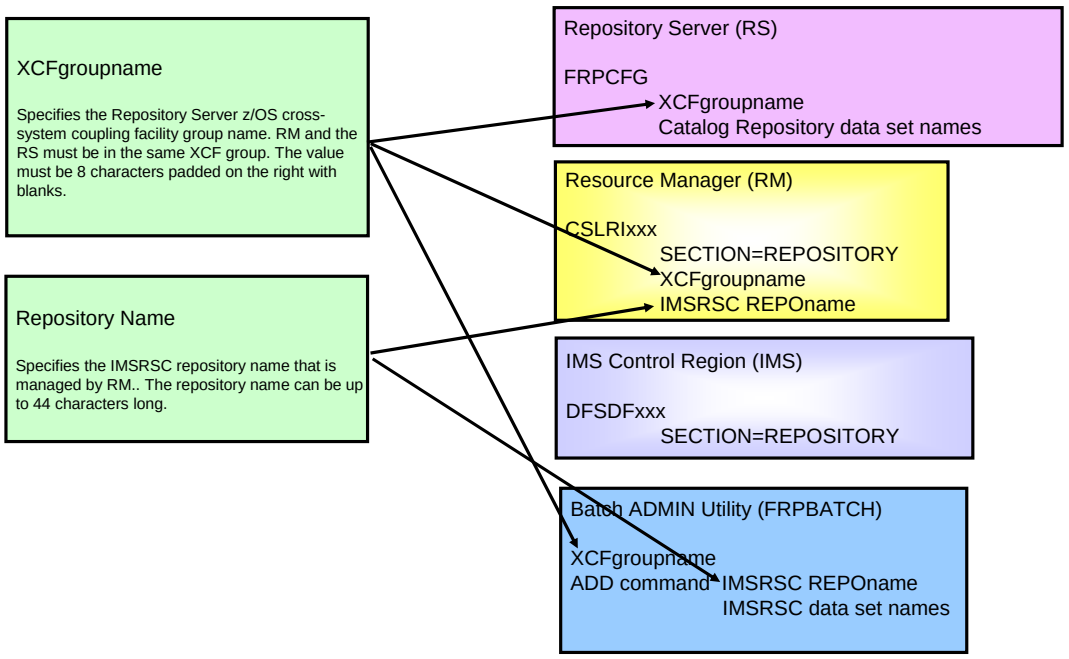
IMS Repository Direct Access (RM not used)

- Repository server commands via 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

There are several repository server commands available to help manage the repository server and IMSRSC repositories. These commands are entered via the z/OS modify interface.

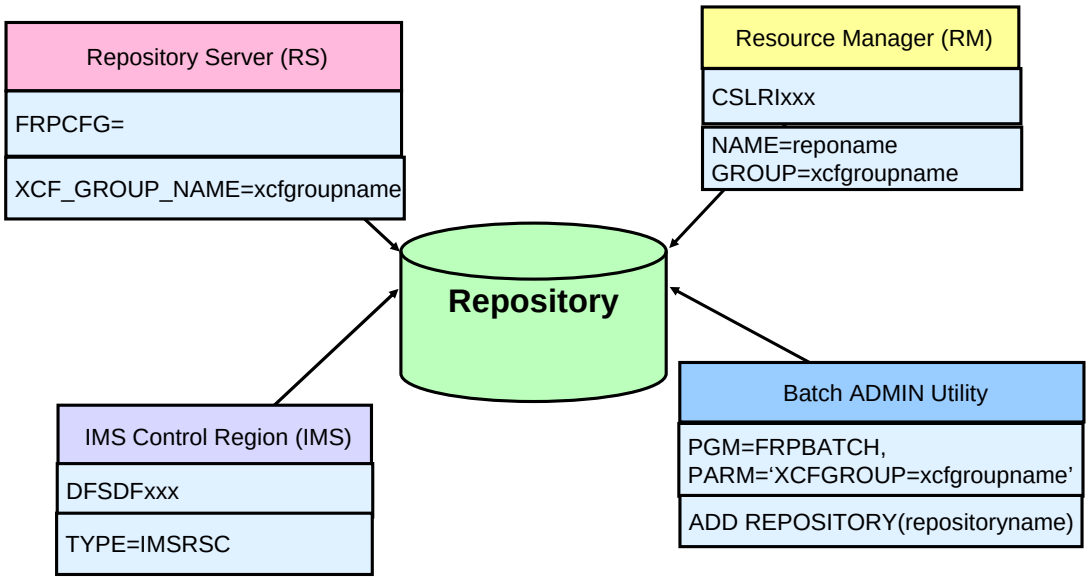
IMS Repository Function Parameter Consistency



137

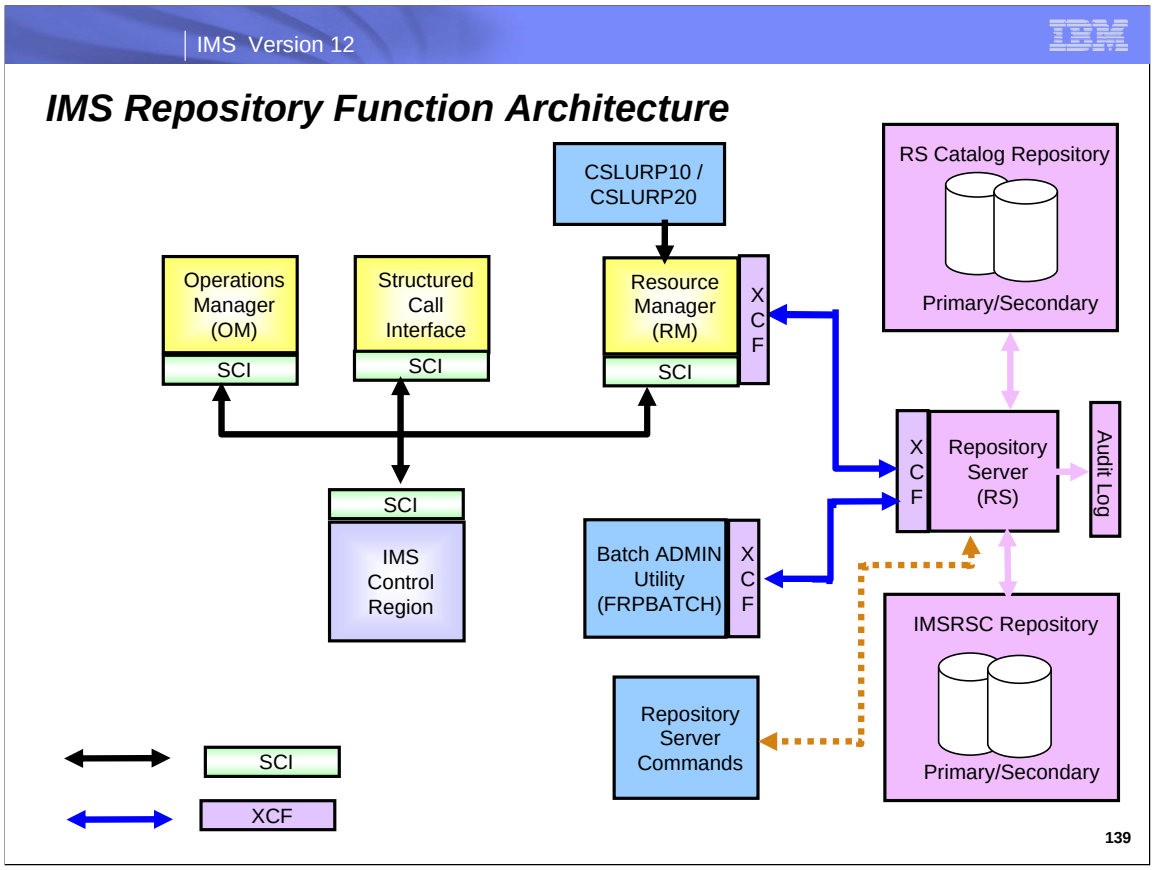
There are two major ‘naming’ parameters that tie together all the processing for a repository server. The first is the z/OS XCFgroupname, which is used for communications to the repository server. The same XCFgroupname must be used in all the member definitions that use a particular repository. The second is the name of the IMSRSC repository which holds the DRD definitions for the IMSplex.

IMS Repository Parameter Consistency Requirements



138

This diagram depicts the parameter settings needed by Repository Server (RS), the Resource Manager (RM), the IMS Control Region (IMS), and the Batch ADMIN Utility that must be identical for access to a particular repository.



This diagram depicts the components of the IMS Repository Function.

The Repository Server, its repositories, and its audit log are in pink. CSL managers are in yellow. Batch and command interfaces are in medium blue.

SCI usage is shown with black arrows. XCF usage is shown with blue arrows.

Appendix

IMS Repository Server Setup / Access

This section covers more details about the major components of the IMS repository function.

Create Catalog Repository Data Sets

- Sample IDCAMS statements for creating the primary catalog repository data sets

```
DEFINE CLUSTER(NAME(IMSPLXA.CAT.REPO1.RID)
REUSE –
INDEXED
KEYS(128,0)
CYLINDERS(1 1)
SHAREOPTIONS(2 3)
FREESPACE(10 10)
RECORDSIZE(282 282)
CONTROLINTERVALSIZE(8192))
DATA(NAME(IMSPLXA.CAT.REPO1.RID.DATA))
INDEX(NAME(IMSPLXA.CAT.REPO1.RID.INDEX))

DEFINE CLUSTER(NAME(IMSPLXA.CAT.REPO1.RMD)
REUSE
INDEXED
KEYS(12,0)
CYLINDERS(1 1)
SHAREOPTIONS(2 3)
FREESPACE(20 20)
RECORDSIZE(8185 8185)
CONTROLINTERVALSIZE(8192))
DATA(NAME(IMSPLXA.CAT.REPO1.RMD.DATA))
INDEX(NAME(IMSPLXA.CAT.REPO1.RMD.INDEX))
```


Here are sample IDCAMS statements for creating the primary catalog repository data sets.

Create Catalog Repository Data Sets

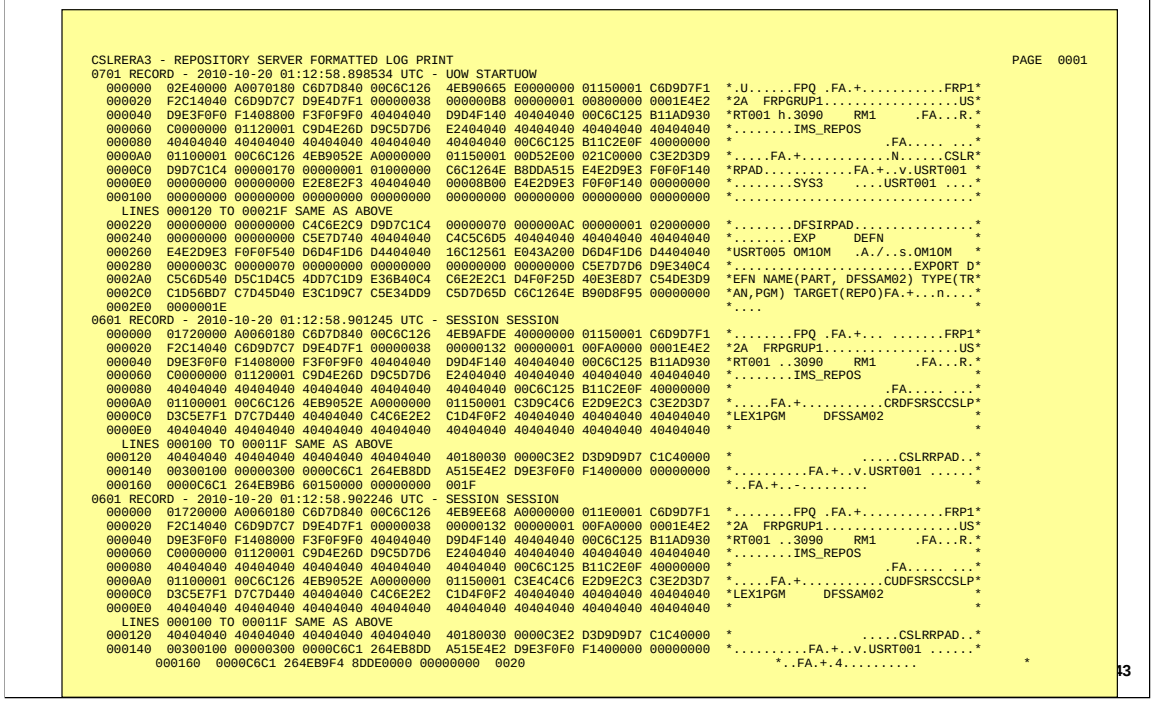
- Sample IDCAMS statements for creating the secondary catalog repository data sets

```
DEFINE CLUSTER(NAME(IMSPLXA.CAT.REPO2.RID)
REUSE
INDEXED
KEYS(128,0)
CYLINDERS(2 1)
SHAREOPTIONS(2 3)
FREESPACE(10 10)
RECORDSIZE(282 282)
CONTROLINTERVALSIZE(8192))
DATA(NAME(IMSPLXA.CAT.REPO2.RID.DATA))
INDEX(NAME(IMSPLXA.CAT.REPO2.RID.INDEX))

DEFINE CLUSTER(NAME(IMSPLXA.CAT.REPO2.RMD)
REUSE
INDEXED
KEYS(12,0)
CYLINDERS(2 1)
SHAREOPTIONS(2 3)
FREESPACE(20 20)
RECORDSIZE(8185 8185)
CONTROLINTERVALSIZE(8192))
DATA(NAME(IMSPLXA.CAT.REPO2.RMD.DATA))
INDEX(NAME(IMSPLXA.CAT.REPO2.RMD.INDEX))
```


Here are sample IDCAMS statements for creating the secondary catalog repository data sets.

Sample Output for Printing the Repository Server Audit Log



```
CSLRERA3 - REPOSITORY SERVER FORMATTED LOG PRINT                                                                PAGE  0001
0701 RECORD - 2010-10-20 01:12:58.898534 UTC - UOW STARTUOW
  000000  02E40000 A0070180 C6D7D840 00C6C126  4EB90665 E0000000 01150001 C6D9D7F1  *.U......FPQ .FA.+...........FRP1*
  000020  F2C14040 C6D9D7C7 D9E4D7F1 00000038  000000B8 00000001 00800000 0001E4E2  *2A  FRPGRUP1.................US*
  000040  D9E3F0F0 F1408800 F3F0F9F0 40404040  D9D4F140 40404040 00C6C125 B11AD930  *RT001 h.3090    RM1     .FA...R.*
  000060  C0000000 01120001 C9D4E26D D9C5D7D6  E2404040 40404040 40404040 40404040  *........IMS_REPOS              *
  000080  40404040 40404040 40404040 40404040  40404040 00C6C125 B11C2E0F 40000000  *                    .FA..... ...*
  0000A0  01100001 00C6C126 4EB9052E A0000000  01150001 00D52E00 021C0000 C3E2D3D9  *.....FA.+...........N......CSLR*
  0000C0  D9D7C1C4 00000170 00000001 01000000  C6C1264E B8DDA515 E4E2D9E3 F0F0F140  *RPAD.....FA.+..v.USRT001 *
  0000E0  00000000 00000000 E2E8E2F3 40404040  00008B00 E4E2D9E3 F0F0F140 00000000  *........SYS3    ....USRT001 ....*
  000100  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000  *................................*
    LINES 000120 TO 00021F SAME AS ABOVE
  000220  00000000 00000000 C4C6E2C9 D9D7C1C4  00000070 000000AC 00000001 02000000  *........DFSIRPAD...............*
  000240  00000000 00000000 C5E7D740 40404040  C4C5C6D5 40404040 40404040 40404040  *........EXP      DEFN          *
  000260  E4E2D9E3 F0F0F540 D6D4F1D6 D4404040  16C12561 E043A200 D6D4F1D6 D4404040  *USRT005 OM1OM   .A./..s.OM1OM   *
  000280  0000003C 00000070 00000000 00000000  00000000 00000000 C5E7D7D6 D9E340C4  *.......................EXPORT D*
  0002A0  C5C6D540 D5C1D4C5 4DD7C1D9 E36B40C4  C6E2E2C1 D4F0F25D 40E3E8D7 C54DE3D9  *EFN NAME(PART, DFSSAM02) TYPE(TR*
  0002C0  C1D56BD7 C7D45D40 E3C1D9C7 C5E34DD9  C5D7D65D C6C1264E B90D8F95 00000000  *AN,PGM) TARGET(REPO)FA.+...n....*
  0002E0  0000001E                                                                  *....                          *
0601 RECORD - 2010-10-20 01:12:58.901245 UTC - SESSION SESSION
  000000  01720000 A0060180 C6D7D840 00C6C126  4EB9AFDE 40000000 01150001 C6D9D7F1  *........FPQ .FA.+... .......FRP1*
  000020  F2C14040 C6D9D7C7 D9E4D7F1 00000038  00000132 00000001 00FA0000 0001E4E2  *2A  FRPGRUP1.................US*
  000040  D9E3F0F0 F1408000 F3F0F9F0 40404040  D9D4F140 40404040 00C6C125 B11AD930  *RT001 ..3090    RM1     .FA...R.*
  000060  C0000000 01120001 C9D4E26D D9C5D7D6  E2404040 40404040 40404040 40404040  *........IMS_REPOS              *
  000080  40404040 40404040 40404040 40404040  40404040 00C6C125 B11C2E0F 40000000  *                    .FA..... ...*
  0000A0  01100001 00C6C126 4EB9052E A0000000  01150001 C3D9C4C6 E2D9E2C3 C3E2D3D7  *.....FA.+...........CRDFSRSCCSLP*
  0000C0  D3C5E7F1 D7C7D440 40404040 C4C6E2E2  C1D4F0F2 40404040 40404040 40404040  *LEX1PGM     DFSSAM02          *
  0000E0  40404040 40404040 40404040 40404040  40404040 40404040 40404040 40404040  *                                *
    LINES 000100 TO 00011F SAME AS ABOVE
  000120  40404040 40404040 40404040 40404040  40180030 0000C3E2 D3D9D9D7 C1C40000  *                .....CSLRRPAD..*
  000140  00300100 00000300 0000C6C1 264EB8DD  A515E4E2 D9E3F0F0 F1400000 00000000  *..........FA.+..v.USRT001 ......*
  000160  0000C6C1 264EB9B6 60150000 00000000  001F                                *..FA.+..-.........            *
0601 RECORD - 2010-10-20 01:12:58.902246 UTC - SESSION SESSION
  000000  01720000 A0060180 C6D7D840 00C6C126  4EB9EE68 A0000000 011E0001 C6D9D7F1  *........FPQ .FA.+...........FRP1*
  000020  F2C14040 C6D9D7C7 D9E4D7F1 00000038  00000132 00000001 00FA0000 0001E4E2  *2A  FRPGRUP1.................US*
  000040  D9E3F0F0 F1408000 F3F0F9F0 40404040  D9D4F140 40404040 00C6C125 B11AD930  *RT001 ..3090    RM1     .FA...R.*
  000060  C0000000 01120001 C9D4E26D D9C5D7D6  E2404040 40404040 40404040 40404040  *........IMS_REPOS              *
  000080  40404040 40404040 40404040 40404040  40404040 00C6C125 B11C2E0F 40000000  *                    .FA..... ...*
  0000A0  01100001 00C6C126 4EB9052E A0000000  01150001 C3E4C4C6 E2D9E2C3 C3E2D3D7  *.....FA.+...........CUDFSRSCCSLP*
  0000C0  D3C5E7F1 D7C7D440 40404040 C4C6E2E2  C1D4F0F2 40404040 40404040 40404040  *LEX1PGM     DFSSAM02          *
  0000E0  40404040 40404040 40404040 40404040  40404040 40404040 40404040 40404040  *                                *
    LINES 000100 TO 00011F SAME AS ABOVE
  000120  40404040 40404040 40404040 40404040  40180030 0000C3E2 D3D9D9D7 C1C40000  *                .....CSLRRPAD..*
  000140  00300100 00000300 0000C6C1 264EB8DD  A515E4E2 D9E3F0F0 F1400000 00000000  *..........FA.+..v.USRT001 ......*
        000160  0000C6C1 264EB9F4 8DDE0000 00000000  0020                                *..FA.+.4.........             *
```

This is page 1 of output from printing the repository audit log after an EXPORT DEFN NAME(PART,DFSSAM02) TYPE(PGM,TRAN) TARGET(REPO) was issued.

Sample Output for Printing the Repository Server Audit Log …

```
0601 RECORD - 2010-10-20 01:12:58.910823 UTC - SESSION SESSION
  000000  01720000 A0060180 C6D7D840 00C6C126  4EBC0671 C0000000 01170001 C6D9D7F1  *........FPQ .FA.+...........FRP1*
  000020  F2C14040 C6D9D7C7 D9E4D7F1 00000038  00000132 00000001 00FA0000 0001E4E2  *2A  FRPGRUP1.................US*
  000040  D9E3F0F0 F1408000 F3F0F9F0 40404040  D9D4F140 40404040 00C6C125 B11AD930  *RT001 ..3090    RM1     .FA...R.*
  000060  C0000000 01120001 C9D4E26D D9C5D7D6  E2404040 40404040 40404040 40404040  *........IMS_REPOS              *
  000080  40404040 40404040 40404040 40404040  40404040 00C6C125 B11C2E0F 40000000  *                    .FA..... ...*
CSLRERA3 - REPOSITORY SERVER FORMATTED LOG PRINT                                                                 PAGE  0020
  0000A0  01100001 00C6C126 4EB9052E A0000000  01150001 C3E4C4C6 E2D3E2E3 C3E2D3D7  *.....FA.+...........CUDFSLSTCSLP*
  0000C0  D3C5E7F1 D9E2C3D3 E3D9C1D5 40404040  E2E8E2F3 40404040 00000000 00000000  *LEX1RSCLTRAN    SYS3    ........*
  0000E0  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000  *................................*
    LINES 000100 TO 00011F SAME AS ABOVE
  000120  00000000 00000000 00000000 00000000  002D0030 0000C3E2 D3D9D9D7 C1C40000  *.....................CSLRRPAD..*
  000140  00300100 00000300 0000C6C1 264EB8DD  A515E4E2 D9E3F0F0 F1400000 00000000  *..........FA.+..v.USRT001 ......*
  000160  0000C6C1 264EBC0D 21970000 00000000  0026                                 *..FA.+...p........              *
0702 RECORD - 2010-10-20 01:12:58.912838 UTC - UOW ENDUOW
  000000  02E40000 A0070280 C6D7D840 00C6C126  4EBC846B 60000000 01170001 C6D9D7F1  *.U......FPQ .FA.+.d,-.......FRP1*
  000020  F2C14040 C6D9D7C7 D9E4D7F1 00000038  000000B8 00000001 00800000 0001E4E2  *2A  FRPGRUP1.................US*
  000040  D9E3F0F0 F1408800 F3F0F9F0 40404040  D9D4F140 40404040 00C6C125 B11AD930  *RT001 h.3090    RM1     .FA...R.*
  000060  C0000000 01120001 C9D4E26D D9C5D7D6  E2404040 40404040 40404040 40404040  *........IMS_REPOS              *
  000080  40404040 40404040 40404040 40404040  40404040 00C6C125 B11C2E0F 40000000  *                    .FA..... ...*
  0000A0  01100001 00C6C126 4EB9052E A0000000  01150001 40C32E00 021C0000 C3E2D3D9  *.....FA.+.......... C......CSLR*
  0000C0  D9D7C1C4 00000170 00000001 01000000  C6C1264E B8DDA515 E4E2D9E3 F0F0F140  *RPAD...........FA.+..v.USRT001 *
  0000E0  00000000 00000000 E2E8E2F3 40404040  00008B00 E4E2D9E3 F0F0F140 00000000  *........SYS3    ....USRT001 ....*
  000100  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000  *................................*
    LINES 000120 TO 00021F SAME AS ABOVE
  000220  00000000 00000000 C4C6E2C9 D9D7C1C4  00000070 000000AC 00000001 02000000  *........DFSIRPAD................*
  000240  00000000 00000000 C5E7D740 40404040  C4C5C6D5 40404040 40404040 40404040  *........EXP     DEFN           *
  000260  E4E2D9E3 F0F0F540 D6D4F1D6 D4404040  16C12561 E043A200 D6D4F1D6 D4404040  *USRT005 OM1OM   .A./..s.OM1OM   *
  000280  0000003C 00000070 00000000 00000000  00000000 00000000 C5E7D7D6 D9E340C4  *.......................EXPORT D*
  0002A0  C5C6D540 D5C1D4C5 4DD7C1D9 E36B40C4  C6E2E2C1 D4F0F25D 40E3E8D7 C54DE3D9  *EFN NAME(PART, DFSSAM02) TYPE(TR*
  0002C0  C1D56BD7 C7D45D40 E3C1D9C7 C5E34DD9  C5D7D65D C6C1264E BC8AC217 00000000  *AN,PGM) TARGET(REPO)FA.+..B.....*
  0002E0  00000027                                                                  *....                           *
0801 RECORD - 2010-10-20 01:12:58.913326 UTC - SYNCPT Start of syncpoint
  000000  00A20000 A0080100 C6D7D840 00C6C126  4EBCA2EB C0000000 01170001 C6D9D7F1  *.s.....FPQ .FA.+.s.........FRP1*
  000020  F2C14040 C6D9D7C7 D9E4D7F1 00000038  00000000 00000001 005A0000 0001E4E2  *2A  FRPGRUP1.............!....US*
  000040  D9E3F0F0 F1408800 F3F0F9F0 40404040  D9D4F140 40404040 00C6C125 B11AD930  *RT001 h.3090    RM1     .FA...R.*
  000060  C0000000 01120001 00C6C125 B11C2E0F  40000000 01100001 00C6C126 4EB9052E  *.........FA..... ........FA.+...*
  000080  A0000000 01150001 0000C6C0 C86E24E2  CA17C6C1 264EBCA9 91D70000 00000000  *.........F.H>.S..FA.+.zjP......*
  0000A0  0028                                                                      *..                             *
0802 RECORD - 2010-10-20 01:12:58.921610 UTC - SYNCPT Syncpoint Phase-1 complete
  000000  00A20000 A0080200 C6D7D840 00C6C126  4EBEA8AF C0000000 010E0001 C6D9D7F1  *.s.....FPQ .FA.+.y.........FRP1*
  000020  F2C14040 C6D9D7C7 D9E4D7F1 00000038  00000000 00000001 005A0000 0001E4E2  *2A  FRPGRUP1.............!....US*
  000040  D9E3F0F0 F1408800 F3F0F9F0 40404040  D9D4F140 40404040 00C6C125 B11AD930  *RT001 h.3090    RM1     .FA...R.*
  000060  C0000000 01120001 00C6C125 B11C2E0F  40000000 01100001 00C6C126 4EB9052E  *.........FA..... ........FA.+...*
  000080  A0000000 01150001 0000C6C0 C86E24E2  CA17C6C1 264EBEB0 DB4E0000 00000000  *.........F.H>.S..FA.+...+......*
```

This is page 20 of output from printing the repository audit log after an EXPORT DEFN NAME(PART,DFSSAM02) TYPE(PGM,TRAN) TARGET(REPO) was issued.

Creating IMS repository data sets

- IMSRSC primary repository data sets (COPY1) sample IDCAMS statements

```
DEFINE CLUSTER(NAME(IMSPLXA.IMSRSC.REPO1.RID)
REUSE -
INDEXED
KEYS(128,0)
CYLINDERS(1 1)
SHAREOPTIONS(2 3)
FREESPACE(10 10)
RECORDSIZE(282 282)
CONTROLINTERVALSIZE(8192))
DATA(NAME(IMSPLXA.IMSRSC.REPO1.RID.DATA))
INDEX(NAME(IMSPLXA.IMSRSC.REPO1.RID.INDEX))

DEFINE CLUSTER(NAME(IMSPLXA.IMSRSC.REPO1.RMD)
REUSE
INDEXED
KEYS(12,0)
CYLINDERS(1 1)
SHAREOPTIONS(2 3)
FREESPACE(20 20)
RECORDSIZE(8185 8185)
CONTROLINTERVALSIZE(8192))
DATA(NAME(IMSPLXA.IMSRSC.REPO1.RMD.DATA))
INDEX(NAME(IMSPLXA.IMSRSC.REPO1.RMD.INDEX))
```

145

Here are sample IDCAMS statements for creating the primary IMSRSC repository data sets.

Creating IMS repository data sets

- IMSRSC secondary repository data sets (COPY2) sample IDCAMS statements

```
DEFINE CLUSTER(NAME(IMSPLXA.IMSRSC.REPO2.RID)
REUSE
INDEXED
KEYS(128,0)
CYLINDERS(2 1)
SHAREOPTIONS(2 3)
FREESPACE(10 10)
RECORDSIZE(282 282)
CONTROLINTERVALSIZE(8192))
DATA(NAME(IMSPLXA.IMSRSC.REPO2.RID.DATA))
INDEX(NAME(IMSPLXA.IMSRSC.REPO2.RID.INDEX))

DEFINE CLUSTER(NAME(IMSPLXA.IMSRSC.REPO2.RMD)
REUSE
INDEXED
KEYS(12,0)
CYLINDERS(2 1)
SHAREOPTIONS(2 3)
FREESPACE(20 20)
RECORDSIZE(8185 8185)
CONTROLINTERVALSIZE(8192))
DATA(NAME(IMSPLXA.IMSRSC.REPO2.RMD.DATA))
INDEX(NAME(IMSPLXA.IMSRSC.REPO2.RMD.INDEX))
```


Here are sample IDCAMS statements for creating the secondary IMSRSC repository data sets.

Creating IMS repository data sets

- IMSRSC spare repository data sets (SPARE) sample IDCAMS statements

```
DEFINE CLUSTER(NAME(IMSPLXA.IMSRSC.REPO3.RID)
REUSE
INDEXED
KEYS(128,0)
CYLINDERS(3 1)
SHAREOPTIONS(2 3)
FREESPACE(10 10)
RECORDSIZE(282 282)
CONTROLINTERVALSIZE(8192))
DATA(NAME(IMSPLXA.IMSRSC.REPO3.RID.DATA))
INDEX(NAME(IMSPLXA.IMSRSC.REPO3.RID.INDEX))

DEFINE CLUSTER(NAME(IMSPLXA.IMSRSC.REPO3.RMD)
REUSE
INDEXED
KEYS(12,0)
CYLINDERS(3 1)
SHAREOPTIONS(2 3)
FREESPACE(20 20)
RECORDSIZE(8185 8185)
CONTROLINTERVALSIZE(8192))
DATA(NAME(IMSPLXA.IMSRSC.REPO3.RMD.DATA))
INDEX(NAME(IMSPLXA.IMSRSC.REPO3.RMD.INDEX))
:
```

147

Here are sample IDCAMS statements for creating the spare IMSRSC repository data sets.

Batch RM utilities

- RDDS to Repository Utility (CSLURP10) sample JCL

```
//RDDS2RPO JOB ,USER,CLASS=A,MSGCLASS=X,NOTIFY=USER
//*
//JOBLIB DD DSN=IMSTESTL.TNUC0,DISP=SHR
//*
//STEP1 EXEC PGM=CSLURP10,MEMLIMIT=4G
//SYSUDUMP DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//*
//*************************************************************/
//* SPECIFY A VALID RDDS DSN FOR INPUT */
//*************************************************************/
//RDDSDSN DD DSN=IMSTESTG.NONSYS.IMSRDDS1,DISP=SHR
//*
//*************************************************************/
//* IMSID MAY BE SPECIFIED ON SYSIN OR DEFAULT TO THE IMSID */
//* ON THE RDDS HEADER RECORD. SUBSTITUTE THE SYSIN STATEMENT */
//* BELOW TO CHANGE BEHAVIOUR */
//*************************************************************/
//SYSIN DD *
IMSPLEX(NAME=PLEX1 IMSID(SYS3,IMS2,IMS3))
//*
```


Here is sample JCL for running the CSLURP10 utility.

Batch RM utilities

- Repository to RDDS Utility (CSLURP20) sample JCL

```
//RPO2DDS JOB ,USER,CLASS=A,MSGCLASS=X,NOTIFY=USER
//*
//JOBLIB DD DSN=IMSTESTL.TNUC0,DISP=SHR
//*
//STEP1 EXEC PGM=CSLURP20
//SYSUDUMP DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//*
//*************************************************************/
//* SPECIFY A VALID NON-SYSTEM RDDS DSN FOR OUTPUT */
//*************************************************************/
//*
//RDDSDSN DD DSN=IMSTESTG.IMS1.NONSYS.IMSRDDS1,DISP=SHR
//*
//*************************************************************/
//* IMSID MUST BE SPECIFIED ON SYSIN. */
//*************************************************************/
//SYSIN DD *
IMSPLEX(NAME=PLEX1 IMSID(IMS1))
//*
```


Here is sample JCL for running the CSLURP20 utility.

IMS Repository Direct Access (RM not used)

- Batch ADMIN utility – FRPBATCH sample JCL

```
//FRPBAT  EXEC PGM=FRPBATCH,PARM='XCFGROUP=FRP2PLEX'
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
    ADD REPOSITORY(IMS_REPOS) +
    REPDSN1RID(IMSPLXA.IMSRSC.REPO1.RID) +
    REPDSN1RMD(IMSPLXA.IMSRSC.REPO1.RMD) +
    REPDSN21RID(IMSPLXA.IMSRSC.REPO2.RID) +
    REPDSN2RRMD(IMSPLXA.IMSRSC.REPO2.RMD) +
    AUTOOPEN(NO)

    START REPOSITORY(IMS_REPOS) MAXWAIT(30,CONTINUE)

    LIST REPOSITORY(IMS_REPOS)
//*
```


Here is a sample of the JCL used to execute the batch ADMIN utility FRPBATCH. The commands are entered as part of SYSIN.