



IMS 10 Dynamic Resource Definition

*Speakers: Janna Mansker, Judy Tse, Alan Smith
IMS Development
Silicon Valley Lab*



Good-Bye IMS SYSGEN for
Databases, Programs, Routing Codes, and Transactions –
Hello IMS Dynamic Resource Definition

February 2009

Dynamic Resource Definition (DRD) Objectives

- **Increase Availability**

- Allow users to dynamically create, update and delete IMS resources without having to perform an IMS cold start or go through the Online Change process
- Allows granular change with minimum impact to availability
- System is quiesced for only those resources to be changed

- **Increase Productivity**

- New User Interfaces
- Simplifies the IMS definition process for IMS systems and resources
- Reduces IMS and z/OS skill set required to define and customize IMS
- Simplifies environment setup for application development

2

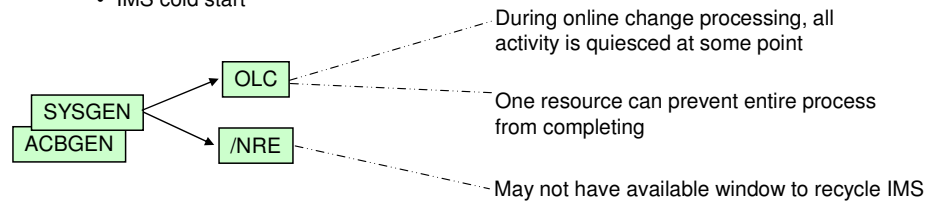
Defining Resources to IMS without DRD

- **To define resources to IMS without DRD**
 - Assembler macro statements must be coded to define the MODBLKS resources (programs, databases, transactions and routing codes) to IMS:
 - APPLCTN, DATABASE, TRANSACT, RTCODE
 - A SYSGEN must then be done to generate the control blocks used to manage those resources
 - PDIR, DDIR, SMB, RCTE
 - The control blocks are saved in the IMS.MODBLKS data set
 - A DBDGEN, PSBGEN and ACBGEN must be done to generate the PSBs and DMBs for the programs and databases
 - The PSB and DMB control blocks are saved in the IMS.ACBLIB data set
 - The MODBLKS control blocks are loaded from the IMS.MODBLKS data set during cold start and used by the online IMS to manage the MODBLKS resources

Modifying Resource Definitions without DRD

- **To add, change, or delete MODBLKS resources in a running IMS system requires**

- MODBLKS SYSGEN
- ACBGEN - if database (DDIR) or program (PDIR) change
- Way to bring the new MODBLKS resources online
 - Online change
 - IMS cold start



- **Process is more complex with multiple IMSs running in an IMSplex**

- Coordinated online change -or- multiple coordinated system restarts

4

February 2009

Prior to DRD: to add, change, or delete a resource required an IMS SYSGEN and either an online change or an IMS restart. Both of these processes resulted in significant (or at least noticeable) unavailability to the end user. Specifically, the online change process results in all (resource) activity being quiesced at some point, including the resources that are not involved in the online change. In addition, one resource that has a condition preventing it from being eligible for online change (such as “work in progress”) can cause an online change to fail, despite the other resources being eligible for inclusion in an online change. Additionally, you might not have an available window of time to shutdown/restart your IMS system. In a multi-IMS plex, it was even more complex because of the need to coordinate the changes (and the unavailability) across multiple IMSs.

Dynamic Resource Definition (DRD)

- **Simplifies resource definition**
- **Allows you to *dynamically* create, update and delete MODBLKS resource definitions without a cold start or online change**
 - *Databases*
 - *Programs*
 - *Transactions*
 - *Routing Codes*
- **Provides increased resource availability**

5

February 2009

The object, therefore, of Dynamic Resource Definition, is to improve the availability of IMS by allowing MODBLKS resource changes without a SYSGEN or online change. The way it does this is by allowing the user to dynamically define and enable MODBLKS resources, which are listed here. These are your databases, programs, transactions and routing codes. When you use DRD, there is no need to perform a IMS SYSGEN, along with a cold start or an online change, when modifying your MODBLKS resource definitions. The availability issues that arise from those methods of managing your MODBLKS resources are eliminated.

DRD Components

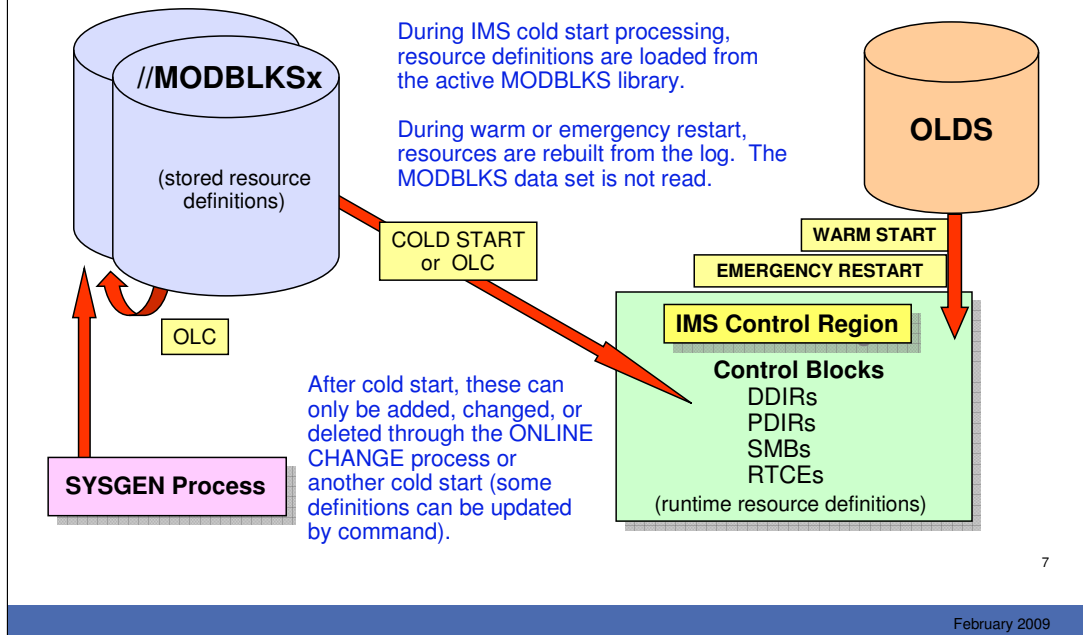
- **Type-2 commands added to dynamically create, update or delete MODBLKS resource definitions**
 - Databases, programs, routing codes and transactions
- **Set of resource definition data sets (RDDS)**
 - To contain resource and descriptor definitions
- **Export and import functions and commands provided**
 - To save/restore resource definitions to/from an RDDS
- **Enhanced Type-2 QUERY commands**
 - To display attributes of defined IMS resources and descriptors
- **Enhanced DFSINSX0 (Destination Creation) user exit**
 - Ability to dynamically create transaction and program resources
- **UPDATE and QUERY support for MSC resources**
 - Supports resource modification without requiring system definition changes or an IMS restart

February 2009

The major components of DRD include:

- new (CREATE, IMPORT, EXPORT) and several commands (UPDATE, DELETE) which are not new but have new functionality
- a set of BSAM data sets called “resource definition data sets” that are used to contain resource definitions that IMS may import from or export to
- an automatic import function that enables IMS to determine the source of its definitions at cold start
- an automatic export function that enables IMS to save new/changed definitions to the RDDS so that the next cold start can import them
- enhancements to the QRY command to show ALL resource and descriptor attributes and status (versus only specific attributes/status)
- an enhanced DFSINSX0 (Destination Creation Exit – renamed from Output Creation Exit) to create program and transactions resources dynamically when a message with an unknown destination arrives, or when the new QUEUE command (also new in IMS 10, which allows the user to queue messages to transactions) is entered with an unknown transaction code
- The ability to update and query MSC resources (no create or delete support)

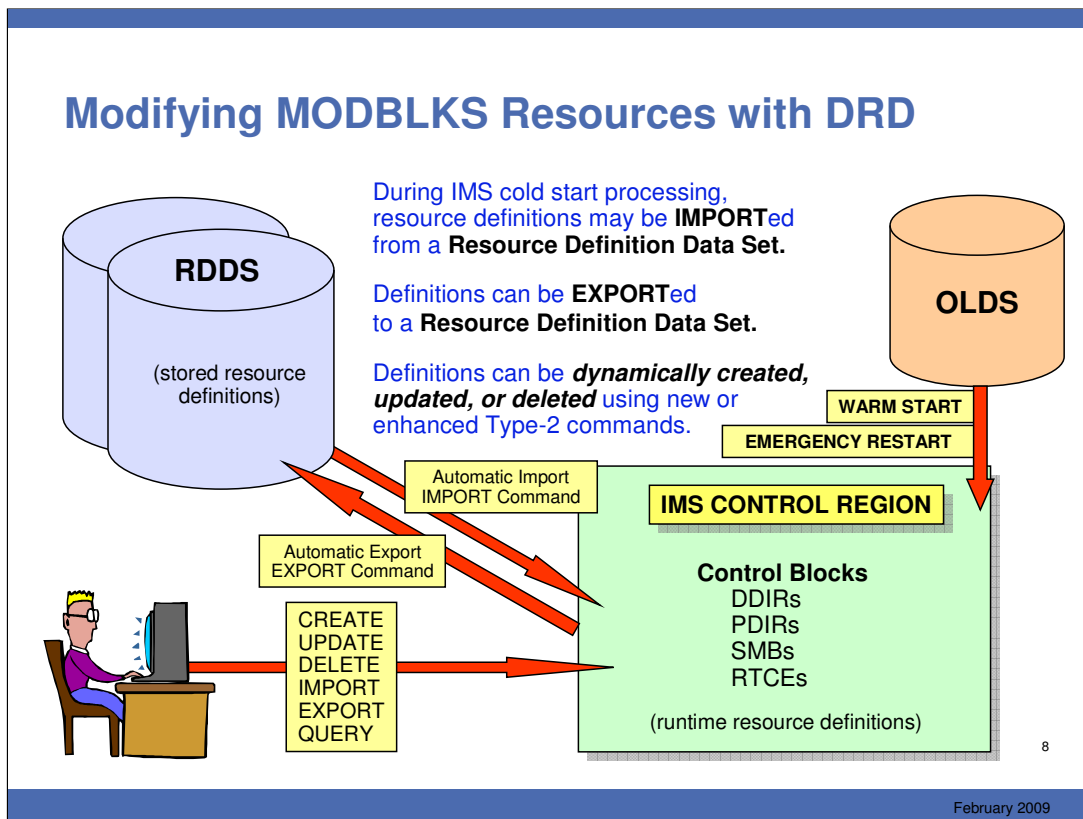
Modifying MODBLKS Resources without DRD



Without DRD, a MODBLKS online change, or a SYSGEN with a cold start is required to update most IMS resources (some attributes could be updated using the /ASSIGN command).

Let's take a look at a diagram showing how you can modify MODBLKS resources without using the DRD capability. The MODBLKS dataset in the upper left corner contains the definitions created during SYSGEN. During an IMS coldstart, these definitions are read into the IMS control region where they are stored as control blocks. At this point, these definitions, or control blocks, can only be modified by doing another SYSGEN plus an online change or IMS restart. Recall the availability issues that arise with either of those methods.

So MODBLKS resources are read from the MODBLKS dataset during IMS coldstart, but where do they come from across a warm or an emergency restart? If you take a look in the upper right corner in this diagram, you can see that they are read from the IMS log, or OLDS (online dataset).



Here is a diagram illustrating how to manage MODBLKS resources using the DRD capability. This diagram is very similar to the one previously shown that depicted how to manage resources without DRD. We now compare/contrast the two diagrams.

The first similarity is the IMS log shown in the upper right corner. When DRD is being used, the MODBLKS resources will still be loaded from the IMS log across warmstart and emergency restart. This does not change when DRD is implemented. MODBLKS resources will be written to the IMS log as x'22' log records as they are created, deleted or updated by the DRD process, as well as at system checkpoint (including initialization and shut down checkpoints). We can see that the RDDS replaces the MODBLKS data set as the primary data set that is used with DRD.

Next, we can see that coldstart has been replaced by the IMPORT function. MODBLKS resources can now be loaded into the IMS control region via either Automatic Import which occurs at IMS coldstart, or they can be loaded when the user issues a Type-2 IMPORT command.

On the other hand, MODBLKS resources can be exported to the RDDS automatically at system checkpoint (depending on the options set for AUTOEXPORT in DFSDFxxx). Additionally, a user can issue the Type-2 EXPORT command to export the resources to the RDDS at a time other than system checkpoint. The exporting of these MODBLKS resources allows them to be saved in an RDDS so that during the next IMS cold start, the resources can be loaded into the IMS control region from the RDDS.

Lastly, we no longer see SYSGEN, online change or IMS restart shown in the diagram. These have been replaced by a user entering Type-2 commands.

DRD Highlights

- **DRD for MODBLKS resources is optional**
- **Either DRD or online change can be enabled for MODBLKS resource management, but not both**
 - Cold start required to switch
- **If DRD is enabled**
 - Online Change for MODBLKS is disabled
 - ACBLIB and FMTLIB Online Change is allowed
 - MODBLKS data sets are optional
 - MODBLKS SYSGEN no longer required

9

February 2009

While using DRD is optional, keep in mind that if DRD is enabled, MODBLKS online change will be disabled. To switch between using DRD and online change to manage MODBLKS resources, a cold start is required.

When DRD is enabled, the MODBLKS data sets that were previously used with MODBLKS online change are no longer required. While MODBLKS online change is disabled, online change for ACBLIB and FMTLIB are still permitted. Lastly, using DRD results in the need for MODBLKS SYSGEN to bring new MODBLKS resources online.

Required Environment for DRD

- **DRD requires IMS 10 with ...**
 - Common Service Layer with SCI and Operations Manager
 - Resource Manager is not required
 - Entry point for DRD commands
 - SPOC, IMS Control Center, Manage Resources Application, other OM interface
 - Data sets to hold resource definitions
 - Resource Definition Data Sets (RDDSs)
- **All IMS online environments supported, including data sharing and shared queues**
 - For applicable resources
- **No longer required**
 - MODBLKS Sysgen process
 - MODBLKS staging, active, and inactive data sets
 - MODBLKS online change process

10

February 2009

DRD is implemented with Type-2 commands and therefore requires the Common Service Layer (CSL) including SCI and OM. RM is not required. In addition, the user must define DRD to IMS by including DRD parameters in the CSL definitions and in the new DFSDfxxx proclib member. RDDS data sets must be defined and of course an entry point for the DRD commands (probably the SPOC or Manage Resources application)

DRD is supported in all IMS environments, including DB/DC, DBCTL, and DCCTL. It is also supported in a data sharing environment and a shared queues environment. Care must be taken in a multi-IMS environment to define resources consistently across all IMSs

When DRD is fully implemented, there is no longer a need for a MODBLKS sysgen, the MODBLKS libraries associated with the sysgen or OLC process, nor is there a need for the MODBLKS DD statements in the IMS online JCL.

Commands Used in Dynamic Resource Definition

- **Type-2 commands entered through OM interface**

Command	Short Form	Purpose
CREATE	CRE	Create resource or descriptor
DELETE	DEL	Delete resource or descriptor
UPDATE	UPD	Update attributes of resource or descriptor Update status of resource
IMPORT	IMP	Import resource or descriptor definitions
EXPORT	EXP	Export resource or descriptor definitions
QUERY	QRY	Query attributes of resource or descriptor Query status of resource

11

February 2009

There are six main commands that are part of DRD.

-CREATE – this is a new command that allows you to define new resources and descriptors to IMS. As part of the create process, the control blocks that are used to manage the resources and descriptors are built.

-DELETE – the delete command has been enhanced to support deleting or resources or descriptors

-UPDATE – the update command has been enhanced to support additional attributes for resources, and to support descriptors.

-IMPORT – this is a new command that allows you to import resource and descriptor definitions from an RDDS. When a resource or descriptor definition is imported, the control blocks used to manage the resource or descriptor are built.

-EXPORT – this is a new command that allows you to export resource and descriptor definitions to an RDDS so that they can be recovered across an IMS cold start.

-QUERY – the query command has been enhanced to show all attributes for resources and descriptors, plus some additional “statuses” that are important to DRD. For example a resource cannot be deleted if it is “in use.”

Descriptors

- **A descriptor is a model for defining (creating) a resource or another descriptor**
 - Establishes *defaults* for attributes not set in the CREATE command
- **IMS-defined descriptors provided with the IMS product:**
 - DFSDS**DB**1 (database)
 - DFSDS**PG**1 (program)
 - DFSDS**TR**1 (transaction)
 - DBFDS**RT**1 (routing code)
- **User-defined descriptors**
 - Created or updated by the user
- **Current default descriptor**
 - Each resource type will have one *default descriptor* (IMS-defined or user-defined with DEFAULT(Y))

12

February 2009

A descriptor is a model or template (you will probably see both terms being used) that can be used by the user to define default attributes for resources being created. You can even use a descriptor as a model in creating another descriptor.

IMS provides four descriptors with the product – one for each resource type. Unless the user creates a descriptor with the DEFAULT(Y) attributes, these IMS descriptors are the “current default descriptors.” The user may create a descriptor which can be specified in the CREATE command or can be set as the current default descriptor, replacing the IMS-defined descriptor. Whichever one is the default is used when the CREATE command does not SET all attributes.

The resource definition process also has default attributes which are used when the user does not completely define all the attributes in the CREATE command. For example, if you do not SET the ACCTYPE attribute in the CREATE DB command, it will default to whatever is in the “default descriptor” – discussed later. Note that the user can set the default attributes to any valid value – he is not bound by what IMS thinks the defaults should be.

Default descriptors are descriptors supplied by IMS or defined by the user which identify what the default attributes should be. There is one default descriptor for each of the 4 resource types. The IMS supplied descriptors have names with the pattern DFSDSxx1, where xx is DB, PG, TR, or RT. The user may define descriptors with any valid attributes.

Recoverability

- **Resource and descriptor definitions**

- Exist for the life of IMS or until deleted
- Are recovered from logs across warm and emergency restart
 - Definitions are logged when created, updated, or deleted, and at system checkpoint time
- Are recovered across cold start if both:
 - Previously exported to RDDS
 - Imported during next cold start

13

February 2009

Resource and descriptor definitions are logged when they are created, updated, or deleted (x'22'), and logged at system checkpoint. IMS warm start restores the checkpointed definitions. IMS emergency restart uses the checkpointed definitions from the restart checkpoint, plus any updates to the end of the log.

If IMS is cold started, then of course it can't use these logs. IMS must import the definitions either from an RDDS (most probable) or you may do a SYSGEN, then cold start IMS and import from MODBLKS. To use the RDDS, it is important that all definitions be previously exported using the AUTOEXPORT option.

Resource Definition Data Sets

- **BSAM data set used to save resource and descriptor definitions**
- **System RDDS**
 - Provides a single system view of an IMS's resources and descriptors
 - Contains all resource and descriptor definitions for an IMS
 - Each IMS must define its own set of system RDDS data sets
 - A set of system RDDS data sets must be defined for Automatic Import and Automatic Export
- **Non-System RDDS**
 - Can be shared between IMSs
 - May contain a subset of an IMS's resource and descriptor definitions

Automatic Export

- **Allows all resource and descriptor definitions for an IMS to be exported to a System RDDS data set automatically at checkpoint time (simple or shutdown)**
- **Enabling Automatic Export is optional**
- **If enabled:**
 - Automatic Export is always done following the restart checkpoint
 - Subsequent exports are only done if one or more definitions have been added, changed or deleted since the last checkpoint
- **A set of system RDDSs must be defined**

15

February 2009

Automatic Import

- **Allows all resource and descriptor definitions to be read in automatically during IMS cold start**

- **Definitions are used to create the internal control blocks needed to manage resources**

- **During cold start, definitions can be imported from**
 - System RDDS
 - MODBLKS data set
 - Or not imported at all

16

February 2009

DFSINSX0 - Destination Creation Exit

- **Enhanced to support the dynamic creation of transactions that can be scheduled**
- **Also enhanced to support the creation of the associated program**
- **Attributes of resources to be created can be**
 - specified by the exit
 - obtained from a model (descriptor or other resource)
 - obtained from the default descriptor
- **Supports creation of transaction on other IMSs, in shared queues environment**

17

February 2009

Manage Resources Application

- **ISPF panel driven application**
 - Support for Dynamic Resource Definition
 - Invoked from IMS Application Menu

- **Panels for managing resources without a knowledge of DRD commands**
 - Actions
 - Create, delete, update, query
 - Resource types supported are those supported by DRD
 - Databases, programs, transactions, routing codes
 - Primary technique is “fill in the blanks”
 - Choices for each parameter are provided
 - Help available for all fields

- **See Reference Section for more information on the Manage Resources Application**

18

February 2009

This function, called “Manage Resources” is invoked from an Application Menu that includes several other programs, including SPOC, KBLA, PDU (HALDB Partition Definition utility).

The program presents a series of panels to simplify the task of creating, deleting, or updating program, transaction, routing code, and database resources. There are also improvements in the QRY function. The primary technique for creating a resource is to present the user with a screen with a field for every attribute which the user can then fill in or let default. The defaults will be shown to the right of the field.

More skilled operators can choose a different format that requires them to know the meaning and valid values for every field, but streamline the process by presenting everything in a single screen.



DRD Commands



Reference Table for Resources & Keywords

Resource Type	SYSGEN Macro	IMS Control Block	Resource Keyword	Descriptor Keyword
Database	DATABASE	DDIR	DB	DBDESC
Program	APPLCTN	PDIR	PGM	PGMDESC
Transaction	TRANSACT	SMB	TRAN	TRANDESC
Routing Code	RTCODE	RTCE	RTC	RTCDESC

- **Examples**

- CREATE PGM...
- DELETE TRANDESC...
- UPDATE TRAN...

20

February 2009

This table shows the resource types, SYSGEN macros, IMS control blocks, and the corresponding command keywords for the resources and descriptors. For example, you might enter the command `CREATE PGMDESC`, `UPDATE DB`, `DELETE TRAN`, or `QRY RTCDESC xxx`. The actual formats for these commands will be shown shortly.

DRD Command Entry and Response

- **Operations Manager (OM)**
 - Commands must be entered through an OM interface
 - OM will consolidate responses from all IMSs in IMSplex
- **If any IMS returns a non-zero return/reason code to OM**
 - OM will return a non-zero return/reason code to SPOC, plus a completion code for each resource
 - If CC not 0, CCText explains reason
 - SPOC displays completion code and text for each IMS and resource
- **Sample command input and response**
 - PGM2 created even though command failed for PGM1

```
CREATE PGM NAME (PGM1 , PGM2) SET (SCHDTYPE (SERIAL) )
```

PgmName	MbrName	CC	CCText
PGM1	IMS1	11	RESOURCE ALREADY EXISTS
PGM2	IMS1	0	

21

DRD Command Entry and Response

- In some cases, an “Error Text” will also be returned
- Sample command input and response

DELETE DB NAME (ACCTMSTR, CUSTMSTR)				
DBName	MbrName	CC	CCText	ErrorText
ACCTMSTR	IMS1	6F	REFERENCED BY PROGRAM	ACCTUPDT
CUSTMSTR	IMS1	0		

- **IMS 10 Command Reference** documents, for each command
 - Return and reason codes
 - CC (Completion Codes)
 - CCText (meaning of completion code – if non-zero)
 - ErrorText (further explanation – if applicable)

22

February 2009

Some command responses may indicate errors that include additional text, called ErrorText. For example, the DELETE DB command successfully deleted CUSTMSTR but could not delete ACCTMSTR because it was “referenced by a program.” The ErrorText in this case indicates that the program referencing the database is ACCTUPDT.

CREATE Command - Syntax

```
CREATE rsc-type | desc-type
NAME (name1, name2, ...)
LIKE (RSC (rsc-name) ) | LIKE (DESC (desc-name) )
SET (attr1 (val1), attr2 (val2), ...)
DEFAULT (Y) <<< valid for descriptors only
```

- **rsc-type | desc-type** = resource or descriptor type
 - DB, DBDESC, PGM, PGMDESC, RTC, RTCDESC, TRAN, TRANDESC
- **NAME()** = resource or descriptor names; they will all have the same attributes; wild cards not supported
- **LIKE()** = resource or descriptor name to use as model
- **SET()** = attribute names and values
- **DEFAULT(Y)** – valid for descriptors only
 - Y causes this descriptor to become current default descriptor

23

February 2009

The slide shows the format of the CREATE command. It is used to create all resources and descriptors.

-rsc-type or desc-type – DB, DBDESC, PGM, PGMDESC, TRAN, TRANDESC, RTC, RTCDESC.

-NAME identifies the name of the definition being created. You can create multiple definitions in a single command. They will all have the same attributes. You cannot use wildcards with the CREATE command.

-SET – this keyword is used to define the resource or descriptor attributes. Any attribute not SET will default to the value in another resource or descriptor identified in the LIKE parameter, or to the value in the current default descriptor for that type.

-LIKE – you can create a resource or descriptor which is LIKE another resource or descriptor. That is, it becomes the “default” for attributes not set. If LIKE is included in the command, then the current system default descriptor is not used for this definition. And, you can create a resource LIKE another resource or descriptor, or a descriptor LIKE another descriptor or resource.

-DEFAULT – this applies only to descriptors. If DEFAULT() is specified, then this descriptor becomes the current system default descriptor. You must only specify one descriptor if Y is coded. If you specify more than one, the command will fail.

CREATE Command - Setting Attributes

- **All attribute values are set either explicitly by SET command keyword or by default**
- **Hierarchy of attribute setting:**
 - SET command keyword – takes precedence over all others
 - LIKE command keyword
 - Attributes not SET are obtained from the specified model
 - Can be LIKE DESC or LIKE RSC
 - Takes precedence over current default descriptor
 - If LIKE command keyword not specified
 - Attributes not SET will be obtained from current default descriptor, which can be either:
 - DFSDSxx1 (IMS-provided)
 - User descriptor created with DEFAULT(Y)
- **See Reference Section for examples**

24

February 2009

SET sets the values for any attribute that you want to be different from what is either in the LIKE definition or the current system default. SET overrides anything. If LIKE is coded, then it sets the values for all descriptors not explicitly SET in the command. If LIKE is not coded, then the current system default descriptor sets the values for any attribute not explicitly SET.

DELETE Command - Syntax

```
DELETE rsc-type | desc-type  
NAME (* | name1, name2, NAME*, ...) OPTION (ALLRSP)
```

- **rsc-type | desc-type** = resource or descriptor type
 - DB, DBDESC, PGM, PGMDDESC, RTC, RTCDESC, TRAN, TRANDESC
- **NAME()** = names of resources or descriptors
 - Can specify multiple names
 - Can use wildcard character
 - NAME(*) – all resources or descriptors (be careful)
 - NAME(PART*) – resource or descriptors starting with PART
 - NAME(ADDPART,BILL*)
- **OPTION(ALLRSP)**
 - Ignored except when NAME(*)
 - Indicates responses to be returned for all resources
 - Default is to return response only for error conditions

25

February 2009

The DELETE command allows the user to delete any of the MODBLKS resources or descriptors. Exceptions include the IMS-defined descriptors, the current system defined default descriptor, the fast path utility program, and the queue control facility (or MRQ) program. These definitions cannot be deleted.

Each resource is deleted individually – some may succeed while others may fail. For those that fail, a condition code describing the reason for the failure will be identified in the response.

This shows the format of the delete command. The command must specify the resource or descriptor type, and one or more names.

The NAME parameter can include multiple names with wild cards, including an “*” which means all definitions of that “type.”

Wild cards are:

- “*” - which means multiple character substitution
- “!” - which is single character substitution (e.g., PA!TMST)
- They may be combined in a single name (e.g., PA!T*)
- They can be used in multiple names (e.g., ADDPART,BILL*,DE!!ART)
- If NAME(*) is specified, OPTION(ALLRSP) indicates that IMS should return a response for every definition deleted. If ALLRSP is not specified, then IMS returns a response only for those definitions for which delete failed.

DELETE Command

- **To delete a resource**
 - Resource cannot be “in use”
 - Recommendation: stop resource before attempting to delete
- **QUERY command enhanced to display resource work status**
 - Show work that might cause delete command (or OLC) to fail
- **See Reference Section for examples**

26

February 2009

Each resource is deleted individually. Individual deletion does not work like online change where either all are deleted or none are deleted. Resources can be successfully deleted if they are not currently in use. If a resource is in use, the delete fails. In a sysplex environment, the delete might succeed on some IMS systems and fail on others. In order to minimize the likelihood that the delete will fail, stop the resource, query the resource to check for work in progress, and complete the work (if any), before attempting the **DELETE** command again.

Examples of resources in use include a database being accessed by an application program, a **/STOP** command or and **UPDATE** command in progress for the resource, a scheduled application program, an active routing code, a transaction with messages queued.

Users can issue UPDATE and DELETE commands without stopping the resource first. The command modules do have code in them to stop the resources if they are not already stopped. Transactions are stopped for queuing and scheduling, programs and databases are stopped for scheduling. Routing codes are also stopped. If the command modules stop a resource, they start the resource again when the command is finished.

The UPDATE and DELETE commands will fail, however, if work is in progress. So, to maximize the likelihood that an UPDATE or DELETE command will succeed, users may want to stop the resource first and make sure all work completes before issuing the UPDATE and DELETE commands. The UPDATE command with the STOP() keyword or the /STOP command can be used to stop the resource. The QUERY command with the SHOW(WORK) filter can be used to check if there is work in progress for the resource. If there is work in progress the user can either wait for the work to finish, or address the work in progress. The steps necessary to complete any work in progress varies depending on the resource type.

UPDATE Command - Syntax

```
UPDATE rsc-type | desc-type  
NAME (name1, name2, ...)  
SET (attr1 (val1), attr2 (val2), ...) |  
START (attr1, attr2, ...) STOP (attr1, attr2, ...)  
DEFAULT (Y) <<< valid for descriptors only
```

- **rsc-type | desc-type** = resource or descriptor type
 - DB, DBDESC, PGM, PGMDESC, RTC, RTCDESC, TRAN, TRANDESC
- **NAME()** = resource or descriptor names; they will all have the same attributes; wild cards are supported
- **SET()** = attribute names and values
- **START()** = function to be started
- **STOP()** = function to be stopped
- **DEFAULT(Y)** – valid for descriptors only
 - Y causes this descriptor to become current default descriptor

27

February 2009

The slide shows the format of the UPDATE command. It is used to update resource and descriptor definitions.

-rsc-type or desc-type – DB, DBDESC, PGM, PGMDESC, TRAN, TRANDESC, RTC, RTCDESC.

-NAME identifies the name of the definition being updated. You can update multiple definitions in a single command. You can use wildcards with the UPDATE command.

-SET – this keyword is used to define the resource or descriptor attributes.

-START – this keyword is used to change the status of a resource. It can be used to start the resource so that it can perform certain function, such as scheduling, or queuing.

-STOP – this keyword is used to change the status of a resource. It can be used to stop the resource so that it no longer can perform a certain function, such as scheduling, or queuing.

-DEFAULT – this applies only to descriptors. If DEFAULT() is specified, then this descriptor becomes the current system default descriptor. You must only specify one descriptor if Y is coded. If you specify more than one, the command will fail.

UPDATE Command

- **Enhanced in IMS 10 to support updating:**
 - Resource/descriptor status and attributes
 - MODBLKS: DB, PGM, TRAN, RTC
 - MSC: MSPLINK, MSLINK, MSNAME
 - MSC definitions have no CREATE | DELETE support
- **Cannot update resource attribute if resource “in use”**
- **Recommendation: QUERY and /STOP resource before attempting to update**
- **Cannot update resource attributes and status in same command**
- **Updating (or creating) descriptor with DEFAULT(Y) sets this descriptor to current system default descriptor for that resource type**
- **See Reference Section for examples**

28

February 2009

The UPDATE command has been enhanced significantly in IMS 10, both to support DRD and to support systems management in general. All of the attributes of resources and descriptors can be “shown” using the QRY command (prior to IMS 10, only a limited number of attributes could be shown, and only for DBs and TRANS).

When an update command is entered with multiple names, each named resource or descriptor is processed individually. Some may succeed and some may fail. The IMS response will identify the reason for any failures with a condition code describing the reason it failed. If entered from the SPOC, the SPOC will identify the completion code (a number) and a description of what the CC means.

For the IMS-defined descriptors, the only valid update is to make the descriptor the default descriptor. This would only be needed if you first made a user-defined descriptor the default and then later wanted to restore DFSDSxx1 to the default.

QUERY Command

- **Enhanced in IMS 10 to support:**
 - QUERY PGM (queries information about program resources)
 - QUERY RTC (queries information about Fast Path routing codes)
 - New parameter SHOW(WORK) displays work status of runtime resource

- **SHOW(WORK) indicates reasons that would prevent online change or some DRD commands from completing successfully**
 - Online Change MODBLKS / ACBLIB
 - DELETE resource
 - UPDATE resource

- **See Reference Section for examples**

29

February 2009

The QUERY command can be issued with the SHOW(WORK) parameter specified to check for any work in progress for the specified resources and other resources that are associated with it. Issuing this command before a DELETE or UPDATE command is useful since it would show you any work in progress that might cause these DELETE or UPDATE commands to fail. The QRY PGM SHOW(WORK) command returns the work status for the program specified. If no work is in progress for the specified resource, a response line is returned with a work status of blanks.

QUERY Command for MSC Resources

- **Enhanced in IMS 10 to display MSC attributes and status**

- QUERY MSPLINK
 - Displays definitions and status for one or more physical links
- QUERY MSLINK
 - Displays definitions and status for one or more logical links
- QUERY MSNAME
 - Displays definitions and status for one or more logical link paths

- **See Reference Section for examples**

30

February 2009

All three MSC definition types can be queried using the type-2 QRY command. The command can be used to display definitional attributes or status.

MSC (Multiple Systems Coupling) is a service of IMS TM that provides the ability to connect geographically dispersed IMS systems. MSC enables programs and operators of one IMS to access programs and operators of the connected IMSs. Communication can occur between 2 or more IMSs (up to 2036) running on any supported combination of operating systems.

IMS 10 introduces the ability to query MSC definitions.

MSPLINK

Displays the physical links that are associated with the specified logical link & logical link path. When SHOW(MSNAME) or SHOW(MSLINK) is specified, STATUS() is invalid. SHOW(MSNAME) and SHOW(MSLINK) are mutually exclusive with all other SHOW parameters.

MSLINK

Displays the logical links that are associated with the specified physical link. When the physical link is VTAM, the maximum number of allowed logical sessions for the physical link is also displayed. When SHOW(MSNAME) is specified, STATUS() is invalid. SHOW(MSNAME) is mutually exclusive with all other SHOW parameters.

MSNAME

Displays the logical link paths that are associated with the specified physical link. The MSNAME names, and the remote and local system identifications are displayed.

EXPORT Command (IMS 10 SPE – PK66704, Oct 08)

- Used to export resource and descriptor definitions to an RDDS
- EXPORT Command Syntax

```
EXPORT DEFN TARGET (RDDS) TYPE () NAME () RDDSDSN () OPTION ()
```

- **Where**
 - **TYPE()** defines the resource type
 - ALL, ALLDESC, ALLRSC ,DB, DBDESC, PGM, PGMDESC, RTC, RTCDESC, TRAN, TRANDESC
 - **NAME()** defines the names of the resources to export
 - NAME(*) is the default
 - **RDDSDSN()** defines a non-system RDDS
 - Cannot be a system RDDS defined in DFSDFxxx member
 - If omitted, exports to a system RDDS

31

February 2009

A new Type-2 command EXPORT was added as an IMS 10 SPE, that allows for the exporting of MODBLKS resources and descriptor definitions to either a system or non-system RDDS. The command syntax is listed in the green box and here is some more detail on each of the parameters:

DEFN: Specifies descriptor and resource definitions.

TARGET(RDDS): Specifies that resources and descriptors are to be exported to an RDDS data set.

TYPE(): Specifies the type of resource or descriptor to export. This is an optional parameter for TARGET (RDDS) and multiple TYPEs can be listed in the same command. The default value is ALL, meaning all resources and descriptors types. Other more granular types that reference a smaller subset of resources/descriptors can be specified as well, for example “all descriptors”, “all resources” and/or individual resource/descriptor types.

NAME(): Specifies the 1-8 character name of a resource or descriptor. For NAME(*) which is also the default value, command responses are returned only for the resource and descriptor names that resulted in an error. OPTION(ALLRSP) can be specified with NAME(*) to obtain the command responses for all of the resource and descriptor names that are processed.

RDDSDSN(dsname): Specifies an optional data set to export resources and descriptors to. This data set cannot be one of the data sets defined on the RDDSDSN= parameter in the DFSDFxxx PROCLIB member (in other words, it cannot be a system RDDS). If the desired target RDDS is the system RDDS, omit this parameter from the command and it will be defaulted to. If there are no system RDDSs defined in an IMS's DFSDFxxx PROCLIB member, then RDDSDSN() is required on the EXPORT command.

EXPORT Command (IMS 10 SPE – PK66704, Oct 08)

▪ EXPORT Command Syntax

```
EXPORT DEFN TARGET (RDDS) TYPE ( ) NAME ( ) RDDSDSN ( ) OPTION ( )
```

– OPTION()

- **OVERWRITE (default)** indicates that resource/descriptor definitions are to overwrite all existing definitions in the RDDS
 - Valid when exporting to system and non-system RDDS
- **APPEND** indicates that resource/descriptor definitions are to be appended to the end of the specified RDDS data set
 - Valid when exporting to non-system RDDS
- **ALLRSP** returns a response for each exported resource, valid with NAME(*) only

32

February 2009

OPTION()

OVERWRITE: Indicates that the resource and descriptor definitions are to overwrite all existing definitions in the RDDS. APPEND is mutually exclusive with OVERWRITE. OVERWRITE is the default and between the two and can be used when exporting to both a system and non-system RDDS.

APPEND: Indicates that the resource and descriptor definitions are to be appended to the end of the specified RDDS data set. If OPTION(NOCHECK) is not specified, validity checking is done to ensure the IMS performing the export is the same IMS that previously exported definitions to the RDDS. The APPEND option can only be used when exporting to a non-system RDDS. APPEND is mutually exclusive with OVERWRITE.

ALLRSP: Indicates that response lines are to be returned for all resources that are processed on the command. The default action is to return response lines only for the resources that resulted in an error. It is only valid when NAME(*) is specified or defaulted to. HALDB partition resources are not exported. OPTION(ALLRSP) does not include these databases in the output and therefore does not flag them with a non-zero completion code.

EXPORT Command (IMS 10 SPE – PK66704, Oct 08)

▪ EXPORT Command Syntax

```
EXPORT DEFN TARGET (RDDS) TYPE ( ) NAME ( ) RDDSDSN ( ) OPTION ( )
```

– OPTION()

- **NOCHECK** valid with the APPEND option and allows the IMS performing the export to be a different IMS than the one that performed the previous export
 - Only valid when exporting to a non-system RDDS

33

February 2009

OPTION()

NOCHECK: Indicates that no ID checking is to be done when definitions are appended to an RDDS. If **OPTION(NOCHECK)** is specified, the IMS performing the export can be a different IMS than the one that performed the previous export. The **NOCHECK** option is only valid when exporting to a non-system RDDS. **NOCHECK** is mutually exclusive with **OVERWRITE** and can only be specified if **APPEND** is specified.

EXPORT Command Restrictions

- **When exporting to a system RDDS**
 - All resource and descriptor definitions must be exported
 - Existing definitions are overwritten
- **When exporting to a non-system RDDS**
 - A subset of resource and descriptor definitions may be exported
 - Existing definitions can be overwritten or appended to
- **IMS defined resources cannot be exported**
 - DBF#FPU0,
 - DBFDSRT1, DFSDSDB1, DFSDSPG1, DFSDSTR1
- **The EXPORT DEFN command can be issued in both a DRD and non-DRD environment**
 - In a non-DRD environment the export must be to a non-system RDDS
- **See Reference Section for examples**

34

IMPORT Command (IMS 10 SPE – PK66682, Oct 08)

- Used to import resource and descriptor definitions from an RDDS
- IMPORT Command Syntax

```
IMPORT DEFN SOURCE(RDDS) TYPE() NAME() RDDSDSN() OPTION()
```

- Where

- **TYPE()** defines the resource type
 - ALL, ALLDESC, ALLRSC, DB, DBDESC, PGM, PGMDESC, RTC, RTCDESC, TRAN, TRANDESC
- **NAME()** defines the names of the resources to import
 - NAME(*) is the default
- **RDDSDSN()** defines a system RDDS or a non-system RDDS
 - If omitted, imports from the current system RDDS.
 - If RDDSDSN() specifies system RDDS, TYPE(ALL) and NAME(*) must be specified, or defaulted to.

35

February 2009

A new Type-2 command IMPORT was added as an IMS 10 SPE, that allows for the importing of MODBLKS resources and descriptor definitions from either a system or non-system RDDS. The command syntax is listed in the green box and here is some more detail on each of the parameters:

DEFN: Specifies descriptor and resource definitions.

SOURCE(RDDS): Specifies where resources and descriptors are to be located for importing. RDDS specifies that resources and descriptors are to be imported from an RDDS.

TYPE(): Specifies the type of resource or descriptor to import. This is an optional parameter for SOURCE (RDDS) and multiple TYPES can be listed in the same command. The default value is ALL, meaning all resources and descriptors types. Other more granular types that reference a smaller subset of resources/descriptors can be specified as well, for example “all descriptors”, “all resources” and/or individual resource/descriptor types.

NAME(): Specifies the 1-8 character name of a resource or descriptor. NAME() supports individual unique names and wildcards for non-system RDDS. For a system RDDS, NAME(*) is the only valid value. For the IMPORT command, NAME(*) is the default value and command responses are returned only for the resource and descriptor names that resulted in an error. OPTION(ALLRSP) can be specified with NAME(*) to obtain the command responses for all of the resource and descriptor names that are processed.

RDDSDSN(dsname): Specifies an optional data set from which to import the descriptor and resource definitions. This data set can be either a system or non-system RDDS. If RDDSDSN() is not specified, the resource and descriptor definitions are imported from the system RDDS that contains the most current definitions (the system RDDSs are defined with the RDDSDSN parameter in the DFSDFxxx PROCLIB member). If RDDSDSN is not specified in the DFSDFxxx PROCLIB member, then RDDSDSN() is required.

IMPORT Command (IMS 10 SPE – PK66682, Oct 08)

▪ IMPORT Command Syntax

```
IMPORT DEFN SOURCE (RDDS) TYPE ( ) NAME ( ) RDDSDSN ( ) OPTION ( )
```

– OPTION()

- **ABORT** imports only if all resources are successful. Otherwise, no resources are to be imported.
- **ALLRSP** returns a response for each imported resource, valid with NAME(*) only

36

February 2009

OPTION(): Specifies additional functions to be performed along with the command.

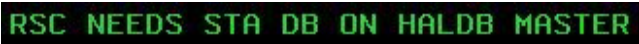
ALLRSP

Indicates that the response lines are to be returned for all resources that are processed on the command. The default action is to return response lines only for the resources that resulted in an error. It is only valid when NAME(*) is specified or defaulted to.

ABORT

Processes the definitions in the data set as a group. If an error occurs in any of the entries, the IMPORT command fails and none of the descriptors or resources are imported.

IMPORT Command Considerations

- **IMPORT supports CREATE of new MODBLKS resources only**
- **IMPORT does not support UPDATE existing MODBLKS resources**
- **HALDB databases: can import HALDB Master only, no partitions**
 - Successful import of HALDB Master will result in ‘ D’ completion code with accompanying text returned:

 - Serves as a reminder that HALDB partitions must be built with either /STA DB command or UPDATE DB START(ACCESS)
- **See Reference Section for examples**

37

February 2009

There are a few restrictions for the IMPORT command. IMPORT can only be used to create new MODBLKS resources. It does not allow for the updating of resources. So for example, PGM1 is updated with the UPDATE command on IMSA and EXPORTed to an RDDS. IMSB cannot IMPORT PGM1 if it already has a PGM1 defined in its online system.

For HALDB databases, IMPORT can only be used to import HALDB Master definitions but not HALDB partitions. HALDB partition definitions are not exported to the RDDS. When a HALDB Master is successfully imported, a completion code of ‘ D’ will be returned with some informational text indicating that the partitions need to be built in order to be brought online. This can be accomplished with either the /START DB command or the UPDATE DB START(ACCESS) command.

Normally, HALDB partitions are read from the RECON during IMS initialization and brought online (control blocks are built) as part of the coldstart process. Since we’re now dynamically importing the HALDB Master, the /START DB or UPDATE DB START(ACCESS) commands need to be used to perform the building of the partitions.

EXPORT/IMPORT with IMS Application Migration

- **Use EXPORT/IMPORT to migrate an application defined on one IMS system to another IMS system**
 - Issue EXPORT command on IMSA to export an application's database, program, routing code and transaction definitions to a non-system RDDS
 - Issue IMPORT command on IMSB to import the definitions

38

February 2009

Using the IMPORT and EXPORT commands can be useful in application migration. These commands can be used to migrate an application's resources from one IMS system to another. For example, the EXPORT command can be issued on IMSA to export the application's resources to a non-system RDDS and IMSB can subsequently import those definitions using the IMPORT command. Note that a non-system RDDS must be used in this case.

EXPORT/IMPORT with IMS Cloning

- **Use EXPORT/IMPORT to port IMS system resources from one IMS system to another IMS system**
- **Example steps**
 1. EXPORT definitions from IMSA to non-system RDDS
 2. Coldstart new IMSB with no resources defined
 3. IMPORT definitions to IMSB from non-system RDDS to port IMSA system resources to IMSB without doing an IMS SYSGEN for databases, programs, routing codes and transactions.

39

February 2009

In the same way, the EXPORT/IMPORT commands can be used in the cloning of IMS systems. For example, the EXPORT command can be used to export IMSA's definitions to a non-system RDDS. IMSB can be coldstarted with no resources defined and can IMPORT IMSA's definitions from the non-system RDDS with the command.



DRD Utilities



New DRD Utilities

- **Projected availability: 1st Quarter 2009**
- **Set of offline utilities to perform the following DRD-related functions:**
 - Generate RDDS from
 - Checkpoint and X'22' (type-2 command) log records
 - MODBLKS data set
 - SYSGEN macros
 - Generate CREATE commands from
 - MODBLKS data set
 - SYSGEN macros
 - Display RDDS content
 - Copy source RDDS to a target RDDS

41

February 2009

In addition to the RDDS Extraction Utility, another set of DRD utilities was created which should be available in 1st quarter 2009. These utilities will also run offline and have functions such as generating an RDDS from either checkpoint and X'22' log records, a MODBLKS data set, or SYSGEN macros. They will also be able to generate CREATE commands from the MODBLKS dataset or from SYSGEN macros. Lastly, as mentioned on the previous chart, the RDDS content will be able to be displayed.

New DRD Utilities

- **Utilities run offline and are invoked via JCL streams**
- **JCL can be generated by either:**
 - ISPF Panels accessible from the Manage Resources application off the IMS Application Menu
 - Utilities will all be documented in the *IMS System Utilities Reference* manual (use of the ISPF panels are not required in order to invoke utilities)

42

February 2009

The DRD utilities are invoked via JCL streams, which can be automatically generated via the Manage Resources application or written using the *IMS System Utilities Reference* as a guide.

Benefits

▪ DRD Utilities

- Facilitate implementation of DRD by reducing manual effort required to create an RDDS
- Make DRD more usable by providing capability to re-create an “up-to-date” RDDS in the event of accidental loss
 - Allow creation of an RDDS from IMS system definitions, MODBLKS data set, checkpoint log records and X'22' log records
 - Allow contents of an RDDS to be queried/displayed/copied

43

February 2009

The DRD Utilities are useful in that they facilitate the implementation of DRD by providing a way to create an RDDS with significantly less effort that would otherwise be necessary. They also make the DRD capability more useable usable by providing a way to re-create an RDDS in the event of an accidentally loss. Using the utilities, an RDDS can be re-created from IMS SYSGEN, MODBLKS, or either checkpoint or X'22' log records. DRD is also made more useable by the utilities providing a way to query the contents of an RDDS.

New DRD Utilities - Example Screenshots

Invoke DRD Utilities via Manage Resource application

```
DFSAPPL          IMS Application Menu
Command ==>
Select an application and press Enter. _

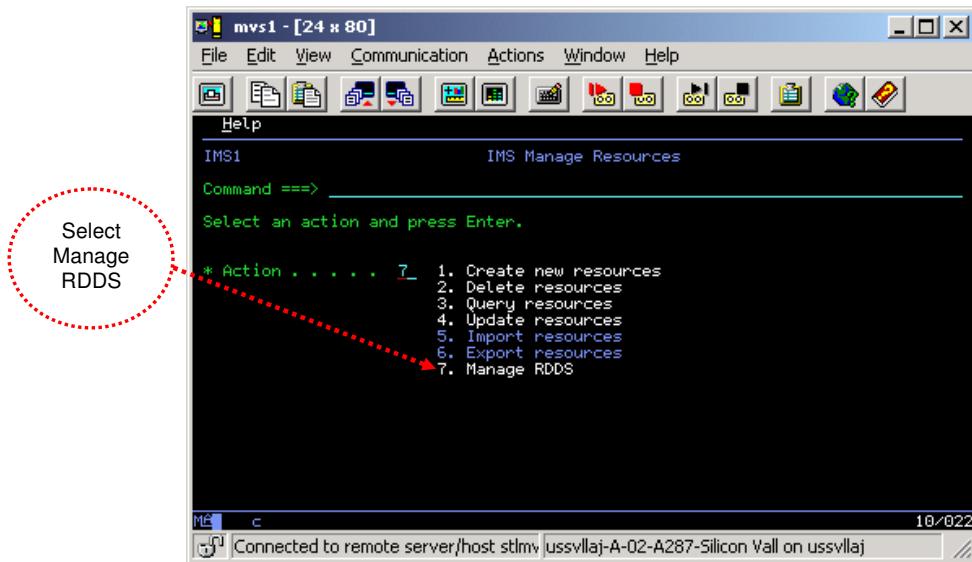
 1 Single Point of Control (SPDC)
 2 Manage resources
 3 Knowledge-Based Log Analysis (KBLA)
 4 HALDB Partition Definition Utility (PDU)
 5 Syntax Checker for IMS parameters (SC)
 6 Installation Verification Program (IVP)
 7 IVP Export Utility (IVPEX)
 8 IPCS with IMS Dump Formatter (IPCS)
 9 Abend Search and Notification (ASN)
10 IBM Support Assistant for IMS (ISA)

To exit the application, press F3.
```

06/041
Connected to remote server/host stlmv ussvllaj-A-02-A287-Silicon Vall on ussvllaj

44

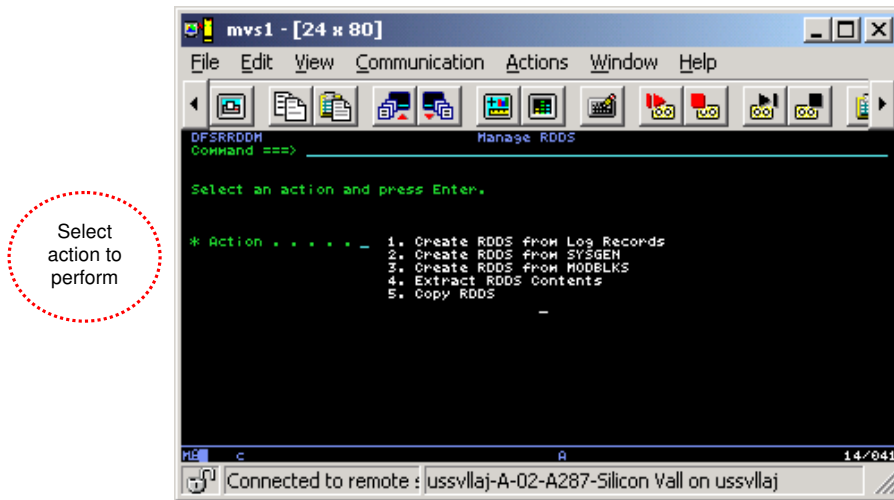
New DRD Utilities - Example Screenshots



45

February 2009

New DRD Utilities - Example Screenshots



46

February 2009

The new DRD utilities also are able to generate CREATE commands from both a MODBLKS data set and from Sysgen macros. However, they are not listed here on this “Manage RDDS” menu. The reason why is that to generate CREATE commands this way, an RDDS first needs to be created and subsequently have its contents extracted with the option of generating Type-2 commands.

Generating an RDDS from log records

```
mvs1 - [24 x 80]
File Edit View Communication Actions Window Help
DFSRDD0 Create RDDS from Log Records
Command ==> More: +

Fill in the following fields and press Enter.

IMS SDFSRESL . . . . 'STLSERV,OPPTST,IMS910,SDFSRESL'
RDDS data set . . . . 'S840536,DFSRDD0,DFSRDDS'

Action . . 1 1. Enter list of log data set names
           2. Extract list of log data set names from RECON

IMS ID . . . . . IMS4
Checkpoint ID . . . . .

Start date/time (UTC) . . . . . (YYYYDD-HHMMSSHH)
Stop date/time (UTC) . . . . . (YYYYDD-HHMMSSHH)

Control data set name . . 'S840536,RDDSWORK,SDFSCNTL'
Work data set HLQ . . . . S840536,DRD
Output space parms: Type . . CYL Primary . . 100 Secondary . . 50
Job JCL statement . . . . 1 1. Use job statement
                           2. Tailor job statement
                           3. Refresh and tailor job statement

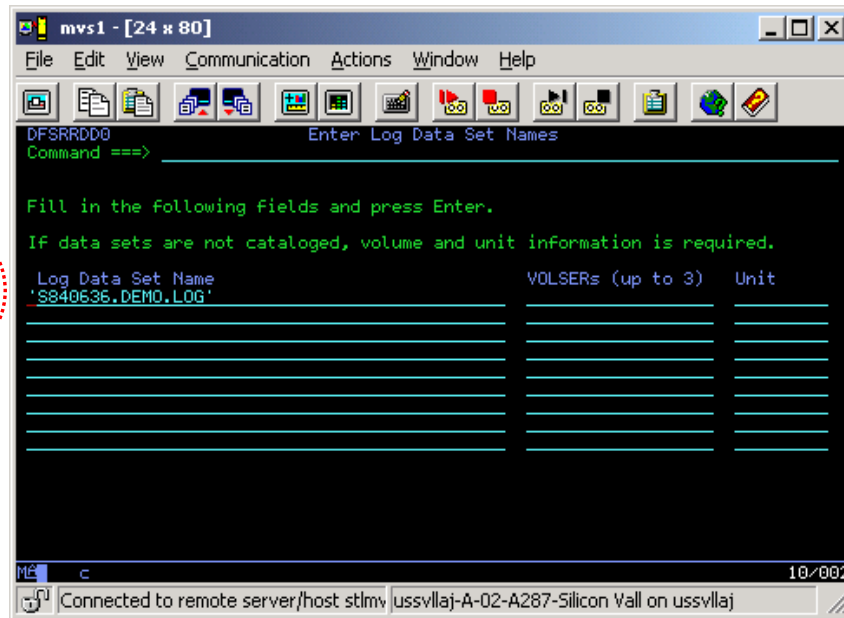
MS 9 13/028
Connected to remote server/host stlmv ussvllaj-A-02-A287-Silicon Vall on ussvllaj
```

47

February 2009

Generating an RDDS from log records

Option 1:
enter log
data set
names



The screenshot shows a terminal window titled 'mvs1 - [24 x 80]' with a menu bar (File, Edit, View, Communication, Actions, Window, Help) and a toolbar. The main display area contains the following text:

```
DFSRRDD0          Enter Log Data Set Names
Command ==> _____

Fill in the following fields and press Enter.
If data sets are not cataloged, volume and unit information is required.
Log Data Set Name          VOLSERs (up to 3)  Unit
'S840636.DEMO.LOG'       _____      _____
_____                   _____      _____
_____                   _____      _____
_____                   _____      _____
_____                   _____      _____
_____                   _____      _____
```

At the bottom of the terminal, there is a status bar showing 'c', '10/002', and 'Connected to remote server/host stlmv ussvllaj-A-02-A287-Silicon Vall on ussvllaj'.

Generating an RDDS from log records

Option 2:
enter
RECON
data set
name

```
mvs1 - [24 x 80]
File Edit View Communication Actions Window Help
DFSRRDDM Extract List of Log Data Sets from RECON
Command ==>

Fill out the following variables and press ENTER.

Dynamic allocation DSN . . . STLSEVU.QPPTEST.IMS910.SDFSRESL
RECON COPY1 DSN. . . . .
RECON COYP2 DSN. . . . .
IMS ID . . . . . SYS3

Log Type . . . 1. OLDS
                2. SLDS
Type '/' to select an option
_ Log is not cataloged. Use unit. .

Start date/time . . . . . - (YYYYDDH-HHMMSS)
Stop date/time . . . . . - (YYYYDDH-HHMMSS)

MS b 10/033
Connected to remote: ussvllaj-A-02-A287-Silicon Vall on ussvllaj
```

Generating an RDDS from log records – JCL generated from panels

```
//job name JOB CLASS=J,MSGCLASS=A,MSGLEVEL=(1,1)
//JOB LIB DD DSN=[library data set name],DISP=SHR
//S1 EXEC PGM=DFSURCL0
//SYSUT1 DD DSN=[Log data set name(s)],DISP=SHR
//RDDSDSN DD DSN=[RDDSDSN data set name],DISP=(,CATLG,DELETE),
// UNIT=SYSDA,VOL=SER=[Volume name],
// SPACE=(CYL,(1,1),RLSE),
// DCB=(LRECL=32756,BLKSIZE=32760,RECFM=VB)
//WORKFILE DD DSN=[Workfile data set name],DISP=(,CATLG,DELETE),
// UNIT=SYSDA,VOL=SER=[Volume name],
// SPACE=(CYL,(1,1),RLSE),
// DCB=(LRECL=133,BLKSIZE=6118,RECFM=FBA)
//REPORT DD SYSOUT=*,
// DCB=(LRECL=133,BLKSIZE=6118,RECFM=FBA)
//CONTROL DD *
IMSID=SYS3
CHKPTID=200809002144935
/*
//
```

50

February 2009

Generating an RDDS from Sysgen – Top of Panel

```
mvs1 - [24 x 80]
File Edit View Communication Actions Window Help
DFSRRDDM Create RDDS from SYSGEN
Command ==>
More: +
Fill in the following fields and press Enter.
IMS SDFSRESL . . . . . STLSEU.FUTGNUC.DRD.SDFSRESL
RDDS data set . . . . . 'S840636.IMS10R.RUPERT.RDDS'
Process . . . 1 1. Perform selections 2-4 below as a single process
                2. Create stage 2 JCL
                3. Create temporary MODBLKS
                4. Process temporary MODBLKS
SYSGEN input DSN. . . . . IMSBLD.IMS10R.STAGE1(C)
SYSGEN copy DSN . . . . . IMSBLD.IMS10R.STAGE1
Modgen DSN. . . . . SYS1.MACLIB
USERLIB DSN . . . . . S840636.RDDSWORK.USERLIB
Object DSN. . . . . S840636.RDDSWORK.OBJDSET
Temp MODBLKS HLQ . . . . . S840636.RDDSWORK.TEMPBLKS
IMS HLQ . . . . . IMSBLD.HMK1010
ASM parameter . . . . . HFLASH
ME f 10/014
Connected to remote : ussvllaj-A-02-A287-Silicon Vall on ussvllaj
```

51

Generating an RDDS from Sysgen – Scrolled down

```
mvs1 - [24 x 80]
File Edit View Communication Actions Window Help
DFSRRDDM Create RDDS from SYSGEN
Command ==>
More: -
SYSGEN copy DSN . . . . . IMSBLD.IMS10R.STAGE1
Modgen DSN. . . . . SY31.MACLIB
USERLIB DSN. . . . . S840636.RDDSWORK.USERLIB
Object DSN. . . . . S840636.RDDSWORK.OBJDSET
Temp MODBLKS HLQ . . . . . S840636.RDDSWORK.TEMPBLKS
IMS HLQ . . . . . IMSBLD.HMK1010
ASM parameter . . . . . HLASM

IMS ID. . . . . IMS1
System type . . . . . DBDC
Suffix . . . . . C

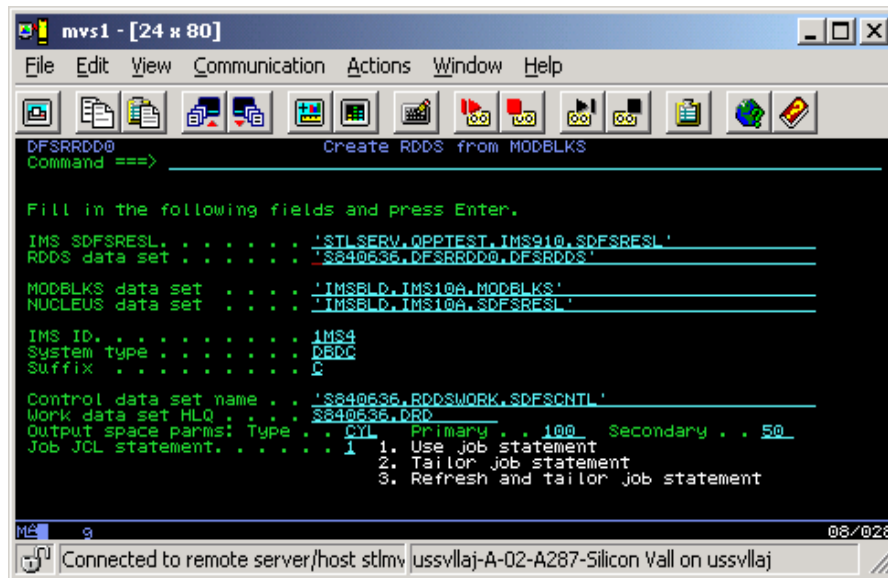
Control data set name . . . S840636.RDDSWORK.SDFSCNTL
Work data set HLQ . . . . . S840636.DRD
Output space parms: Type . . . CYL Primary . . 100 Secondary . . 50
RDDS Data set volume . . . . . (Optional)
Job JCL statement. . . . . 1 1. Use job statement
                             2. Tailor job statement
                             3. Refresh and tailor job statement

ME f 04/014
Connected to remote: ussvllaj-A-02-A287-Silicon Vall on ussvllaj
```

52

February 2009

Generating an RDDS from MODBLKS



```
mvs1 - [24 x 80]
File Edit View Communication Actions Window Help
DFSRRDD0 Create RDDS from MODBLKS
Command ==>

Fill in the following fields and press Enter.

IMS SDFSRESL . . . . . 'STL.SRW.QPPTEST.IMS910.SDFSRESL'
RDDS data set . . . . . 'S840636.DFSRRDD0.DFSRDD0'

MODBLKS data set . . . . . 'IMSBLD.IMS10A.MODELKS'
NUCLEUS data set . . . . . 'IMSBLD.IMS10A.SDFSRESL'

IMS ID . . . . . IMS4
System type . . . . . DBDC
Suffix . . . . . 4

Control data set name . . 'S840636.RDDSWORK.SDFSCNTL'
Work data set HLQ . . . . 'S840636.DBP'
Output space parms: Type . . CYL Primary . . 100 Secondary . . 50
Job JCL statement . . . . 1 1. Use job statement
                          2. Tailor job statement
                          3. Refresh and tailor job statement

MS 9 08/028
Connected to remote server/host stlmv ussvllaj-A-02-A287-Silicon Vall on ussvllaj
```

53

February 2009

Generating an RDDS from MODBLKS – JCL generated from panels

```
job name JOB CLASS=J,MSGCLASS=A,MSGLEVEL=(1,1)
//JOBLIB DD DSN=[library data set name],DISP=SHR
//S1 EXEC PGM=DFSURCM0
//MODBLKS DD DSN=[MODBLKS data set name],DISP=SHR
//RDDSDSN DD DSN=[RDDSDSN data set name],DISP=(,CATLG,DELETE),
// UNIT=SYSDA,VOL=SER=[Volume name],
// SPACE=(CYL,(1,1),RLSE),
// DCB=(LRECL=32756,BLKSIZE=32760,RECFM=VB)
//SYSPRINT DD SYSOUT=*,
// DCB=(LRECL=133,BLKSIZE=6118,RECFM=FBA)
//REPORT DD SYSOUT=*,
// DCB=(LRECL=133,BLKSIZE=6118,RECFM=FBA)
//CONTROL DD *
IMSID=SYS3
SUF=V
/*
```

54

February 2009

RDDS Extraction Utility (DFSURDD0)

- **Offline batch utility**
- **Converts stored resource definitions in a system or non-system RDDS to:**
 - IMS Stage-1 macro statements (APPLCTN, TRANSACT, DATABASE, RTCODE)
 - IMS Type-2 CREATE commands for DRD (could be submitted to IMS via batch SPOC)
 - CREATE DB
 - CREATE TRAN
 - CREATE PGM
 - CREATE RTC
 - RDDS Content query and statistics report
- **Queries the RDDS and displays the resources/descriptors with all attribute values, also reports duplicates (available 1st quarter 2009)**
- **Output written to SYSPRINT data set**

55

February 2009

The RDDS Extraction Utility (known as DFSURDD0) runs offline and reads the RDDS to convert the contents into either Stage 1 macro statements or IMS Type-2 CREATE commands. It can also query the RDDS to display its contents, which will be resources and descriptor definitions along with their attribute values. Duplicates are flagged as well. When this utility is run, the output is written to the SYSPRINT data set. This utility is available now, but the query RDDS function will be delivered in 4th quarter 2008 as an SPE with some other new DRD utilities, which we'll cover now.

RDDS Extraction Utility- Example Screenshot

```
mvs1 - [24 x 80]
File Edit View Communication Actions Window Help
DFSRRDD0 Extract RDDS Contents Top of data
Command ==>
Fill in the following fields and press Enter.
IMS SDFSRESL . . . . . 'STLSERV.QPPTEST.IMS910.SDFSRESL'
RDDS data set . . . . . 'S840636.DFSRRDD0.DFSRDDS'
Process Selection
Type '/' to select an option
Generate stage 1 macro statements
Output data set . . . . . 'S840636.RDDSWORK.STAGE1'
Generate CREATE statements
Output data set . . . . . 'S840636.RDDSWORK.CMDATE'
Query RDDS contents
Output data set . . . . . 'S840636.RDDSWORK.QUERY'
Control data set name . . . . . 'S840636.RDDSWORK.SDFSCNTL'
Work data set HLQ . . . . . S840636.DRD
Output space parms: Type . . . . . CYL Primary . . 100 Secondary . . 50
Job JCL statement . . . . . 1 1. Use job statement
2. Tailor job statement
3. Refresh and tailor job statement
c 08/032
Connected to remote server/host stlmv ussvllaj-A-02-A287-Silicon Vall on ussvllaj
```

56

February 2009

RDDS Extraction Utility – JCL Generated

```
//job name JOB CLASS=J,MSGCLASS=A,MSGLEVEL=(1,1)
//JOB LIB DD DSN=[library data set name],DISP=SHR
//S1 EXEC PGM=DFSURDD0,MEMLIMIT=4G
//RDDSDSN DD DSN=[RDDS data set name],DISP=SHR
//SYSOUT DD DSN=[output data set name],DISP=(,CATLG,DELETE),
// UNIT=SYSDA,VOL=SER=[Volume name],
// SPACE=(CYL,(1,1),RLSE),
// DCB=(LRECL=80,RECFM=FB,BLKSIZE=800)
//SYSPRINT DD SYSOUT=*
//CONTROL DD *
OUTPUT=QUERY ←
/*
//
```

This example shows
the JCL for querying
the RDDS contents

57

February 2009

RDDS Copy RDDS Utility- Example Screenshot

```
mvs1 - [24 x 80]
File Edit View Communication Actions Window Help
DFSRRDDM COPY RDDS
Command ==>
Fill in the following fields and press Enter.
IMS SDFSRESL . . . . . SILSERV.QPPTEST.IMS310.SDFSRESL
Source RDDS data set . . . . . S340636.MOORELKS.RDDS
Target RDDS data set . . . . . S340636.TEST.RDDS.NORMAN
IMS ID. . . . . IMS1
Type \'/\' to select an option
Retain RDDS header timestamp from Source RDDS?
Control data set name . . . . . S340636.RDDSWORK.SDFSCHTL
Work data set HLQ . . . . . S340636.RDD
Output space parms: Type . . . . . QYL PRIMARY . . 1 Secondary . . 5
Job JCL statement . . . . . 1
1. Use job statement
2. Tailor job statement
3. Refresh and tailor job statement
ME c 09/033
Connected to remote : ussvllaj-A-02-A287-Silicon Vall on ussvllaj
```

58

February 2009

RDDS Copy Utility – JCL Generated

```
//job name JOB CLASS=J,MSGCLASS=A,MSGLEVEL=(1,1)
//JOBLIB DD DSN=[library data set name],DISP=SHR
//S1 EXEC PGM=DFSURCP0
//RDDSIN DD DSN=[Source RDDS data set name],DISP=SHR
//RDDSdsn DD DSN=[Target RDDS data set name],
// DISP=(,CATLG,DELETE),
// UNIT=SYSDA,VOL=SER=[Volume name],
// SPACE=(CYL,(15,1),RLSE),
// DCB=(LRECL=32756,BLKSIZE=32760,RECFM=FB)
//REPORT DD SYSOUT=*,DCB=(LRECL=133,BLKSIZE=6118,RECFM=(FBA)
//CONTROL DD *
IMSID=IMS1
RETAINTIME
/*
//
```

59

February 2009

Summary



Summary

▪ **IMS 10 Dynamic Resource Definition**

- Improves IMS availability by allowing dynamic creation, deletion, and updating of MODBLKS runtime resource definitions
 - Reduces the requirements for planned outages
 - Eliminates unavailability associated with MODBLKS OLC and IMS restart
- MSC DRD allows dynamic updating of MSC definitions

▪ **DRD Utilities**

- Allow user to generate DRD-related components using existing resources
 - RDDS
 - IMS SYSGEN macro statements
 - Type-2 CREATE commands
- Allow user to query/display/copy RDDS contents

▪ **See following Reference Section**

- Examples of DRD command input/output
- Manage Resources application (optional interface for using DRD)

61



Reference Section





DRD Command Examples



CREATE Command - Examples

- Creating a transaction with the IMS default descriptor

```
CRE TRAN NAME (BALCHECK) SET (PGM (ACCTPGM) )
```

- Default descriptor is used since LIKE not specified
 - Could be IMS defined descriptor, DFSDSTR1
 - Or user defined descriptor
- All attribute values besides PGM are obtained from default descriptor

- Creating a transaction with a user-defined descriptor

```
CRE TRANDESC NAME (ACCTMODL)  
      SET (PARLIM (5) , PLCT (20) , PLCTTIME (2) , MAXRGN (5) )
```

```
CRE TRAN NAME (ACCTUPDT) LIKE (DESC (ACCTMODL)  
      SET (MAXRGN (8) , PGM (ACCTPGM) )
```

```
CRE TRAN NAME (ACCTDLET) LIKE (RSC (ACCTUPDT) )
```

64

DELETE Command - Examples

- Examples of deleting MODBLKS runtime resource definitions

```
QRY PGM NAME (ACCTPGM) SHOW (WORK)
UPD PGM NAME (ACCTPGM) STOP (SCHD)
DEL PGM NAME (ACCTPGM)
```

```
QRY TRAN NAME (ACCTDLET, ACCTADD, ACCTU*) SHOW (WORK)
UPD TRAN NAME (*) STOP (SCHD)
DEL TRAN NAME (ACCTDLET, ACCTADD, ACCTU*)
```

UPDATE Command - Examples

- Examples of updating MODBLKS runtime resource definitions

```
UPD DB NAME (ACCTMSTR) START (ACCESS) SET (ACCTYPE (EXCL) )
```

```
UPD PGM NAME (ACCT*) SET (RESIDENT (Y) )
```

```
UPD TRAN NAME (ACCTUPDT) SET (PARLIM (5) , RESP (Y) )
```

```
UPD TRANDESC NAME (COMPMODL) SET (MAXRGN (10) , DEFAULT (Y) )
```

UPDATE Command for MSC - Examples

- Activating previously defined “dummy” definitions between IMSA and IMSB
- On IMSA

```
UPD MSPLINK NAME (DMYMSP1)
      SET (MSPLINK (IMSAB) , NODE (IMSB) )

UPD MSLINK NAME (DMYMSL1)
      SET (MSLINK (IMSAB1) , MSPLINK (IMSAB) )

UPD MSNAME NAME (DMYMSN1)
      SET (MSNAME (LINKA1) , (SIDR (30) , SIDL (20) )
```

- Repeat for IMSB

67

QUERY Command - Examples

- Examples of how to determine if there is work in progress for a resource or if the resource is referenced by another resource

```
QRY DB NAME (ACCTMSTR) SHOW (WORK | PGM)
```

```
QRY PGM NAME (ACCTPGM) SHOW (WORK | TRAN | RTC)
```

```
QRY TRAN NAME (ACCT*) SHOW (WORK)
```

```
QRY RTC NAME (ACCT*) SHOW (WORK | REGION)
```

68

February 2009

Here are some QUERY examples. This first QRY DB command would show any work in progress for the database along with all of the programs that reference it. The next QUERY PGM command would output any work in progress for this program AND any transactions or routing codes that are associated with this program.

Any work in progress that is output from these commands could prevent the following from executing successfully: MODBLKS or ACBLIB online change....as well as the DRD commands CREATE/UPDATE/DELETE.

QUERY Command for MSC Resources - Examples

- Examples of querying MSC resource definitions

```
QUERY MSPLINK NAME (IMSAB)
      SHOW (MSLINK, MSNAME)

QUERY MSLINK NAME (IMSAB1)
      SHOW (PARTNER, MSPLINK)

QUERY MSNAME NAME (LINKA1)
      SHOW (MSLINK, MSPLINK, SYSID)
```

EXPORT Command - Example 1

TSO SPOC Input

```
EXPORT DEFN TARGET (RDDS)
```

TSO SPOC Output Response

```
IMSpdex . . . . . : PLEX1
Routing . . . . . :
Start time. . . . . : 2008.294 15:55:10.77
Stop time . . . . . : 2008.294 15:55:10.82
Return code . . . . : 00000000
Reason code . . . . : 00000000
Reason text . . . . :
Command master. . . : IMS1

MbrName  Messages
IMS1     DFS3404I EXPORT COMMAND COMPLETE FOR ALL ,
DSN=IMS.SYSTEM.RDDS1
```

70

February 2009

EXPORT Command - Example 2

TSO SPOC Input

```
EXPORT DEFN TARGET(RDDS) RDDSDSN(NON.SYS.RDDS1)
NAME(D*, E*, F*,HIMALM01,TPDYNCUR) TYPE(ALL)
OPTION(OVERWRITE)
```

TSO SPOC Output Response

Name	Type	MbrName	CC
DBFSAMD2	DB	IMS1	0
EMHTX2	TRAN	IMS1	0
EMHTX3	TRAN	IMS1	0
FESTXA	TRAN	IMS1	0
FESTXB	TRAN	IMS1	0
HIMALM01	PGM	IMS1	0
TPDYNCUR	PGM	IMS1	0

71

February 2009

EXPORT Command - Example 2

If you hit PF4 you see the following response:

```
IMSpIex . . . . . : PLEX1
Routing . . . . . :
Start time. . . . . : 2008.294 16:05:10.77
Stop time . . . . . : 2008.294 16:05:10.82
Return code . . . . : 00000000
Reason code . . . . : 00000000
Reason text . . . . :
Command master. . . : IMS1

MbrName  Messages
  IMS1    DFS3404I EXPORT COMMAND COMPLETE FOR ALL ,
DSN=NON.SYS.RDDS1
```

72

February 2009

EXPORT Command - Example 3

TSO SPOC Input

```
EXPORT DEFN TARGET (RDDS) RDDSDSN (NON.SYS.RDDS1)  
TYPE (ALLRSC) NAME (TEST)
```

TSO SPOC Output Response

Name	Type	MbrName	CC
TEST	PGM	IMS1	0

EXPORT Command - Example 4

TSO SPOC Input

```
EXPORT DEFN TARGET (RDDS) RDDSDSN (NON.SYS.RDDS2)  
NAME (TEST, PGMTES*)
```

TSO SPOC Output Response

Name	Type	MbrName	CC
TEST	PGM	IMS1	0
PGMTEST	PGMDESC	IMS1	0
TEST	TRANDESC	IMS1	0

IMPORT Command - Example 1

TSO SPOC Input

```
IMPORT DEFN SOURCE (RDDS) OPTION (ABORT, ALLRSP)
```

TSO SPOC Output Response

```
IMSpIex . . . . . : PLEX1
Routing . . . . . :
Start time. . . . . : 2008.294 15:55:04.40
Stop time . . . . . : 2008.294 15:55:04.41
Return code . . . . : 0200000C
Reason code . . . . : 00003004
Reason text . . . . : No requests were successful.
Command master. . . : IMS1

      Return      Reason
MbrName   Code      Code      Description
-----
IMS1      00000008    0000211F    No system RDDSs
defined, rddsdn() required
```

75

IMPORT Command - Example 2

TSO SPOC Input

```
IMPORT DEFN SOURCE(RDDS) RDDSDSN(NON.SYS.RDDS1)
```

TSO SPOC Output Response

```
IMSpIex . . . . . : PLEX1
Routing . . . . . :
Start time . . . . : 2008.294 15:55:05.36
Stop time . . . . . : 2008.294 15:55:07.45
Return code . . . . : 00000000
Reason code . . . . : 00000000
Reason text . . . . :
Command master . . : IMS1
MbrName Messages
-----
IMS1      DFS3405I IMPORT COMMAND COMPLETE FOR ALL ,
DSN=NON.SYS.RDDS1
```

76

IMPORT Command - Example 3

TSO SPOC Input

```
IMPORT DEFN SOURCE (RDDS) RDDSDSN (IMSPLEX1.RDDS.DEFN)  
OPTION (ALLRSP)
```

TSO SPOC Output Response

Name	Type	MbrName	CC	ImpType
DEDBJN01	DB	IMS2	0	CREATE
DBDFLT	DBDESC	IMS2	0	CREATE
CDEBS	PGM	IMS2	0	CREATE
PGMDFLT	PGMDESC	IMS2	0	CREATE
SMQFP5A	RTC	IMS2	0	CREATE
RTCDFLT	RTCDESC	IMS2	0	CREATE
CDEBTRN1	TRAN	IMS2	0	CREATE
TRANDFLT	TRANDESC	IMS2	0	CREATE

77

IMPORT Command - Example 4

TSO SPOC Input

```
IMPORT DEFN SOURCE (RDDS) RDDSdsn (MYDSN.DEFN)  
OPTION (ABORT, ALLRSP)
```

TSO SPOC Output Response

Name	Type	MbrName	CC	CCText
PART	DB	IMS1	9	NO IMPORT - OPTION ABORT
PARTMAST	DB	IMS1	9	NO IMPORT - OPTION ABORT
TESTDB	DB	IMS1	9	NO IMPORT - OPTION ABORT
TRND01	TRAN	IMS1	6D	INVALID PROGRAM NAME
TRND11	TRAN	IMS1	9	NO IMPORT - OPTION ABORT
TRND123	TRAN	IMS1	9	NO IMPORT - OPTION ABORT
TRND01	TRANDESC	IMS1	9	NO IMPORT - OPTION ABORT

78

February 2009



Manage Resources Application



Manage Resources (MR) - Overview

- What is “Manage Resources”?
 - IMS application which guides the user through the DRD process
 - Invoked from IMS Application Menu
 - Includes a series of panels to let the user perform DRD functions against IMS MODBLKS resources and descriptors
- What kind of actions can you perform with MR?
 - Create, delete, update, query, manage RDDS
- Resource types supported are those supported by DRD
 - Databases, programs, transactions, routing codes

80

February 2009

MR is a new enhancement of the DFSSPOC program. It greatly reduces the need for the user to know the names of all the attributes and their valid values. It also eliminates the need to know the exact format of the commands. For those that do know the names and values for the attributes, a Command Syntax View is available to reduce the number of screens required for some commands – especially those related to transactions which have lots of attributes.

This function, called “Manage Resources” or MR as I use in this presentation for brevity, is invoked from an Application Menu that includes several other programs, including SPOC, KBLA, PDU, and several others.

The program presents a series of panels to simplify the task of creating, deleting, or updating MODBLKS resources. There are also improvements in the QRY function.

Manage Resources (MR) - Overview

- How do you use MR?
 - Primary technique is “fill in the blanks”
 - Panels and navigation similar to TSO SPOC panels

- Supports various skill levels
 - Graduated levels of help available
 - Fewer screens required for higher skill levels

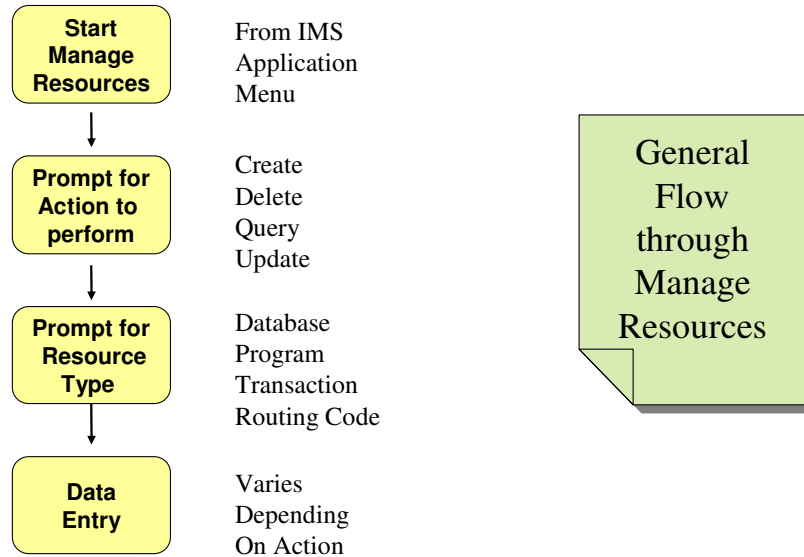
81

February 2009

The primary technique for creating (for example) a resource is to present the user with a screen with a field for every attribute which the user can then fill in or let default. The defaults will be shown to the right of the field.

More skilled operators can choose a different format that requires them to know the meaning and valid values for every field, but streamline the process by presenting everything in a single screen.

Hierarchical Panel Structure



82

February 2009



The basic flow through MR is:

- Start MR from the IMS Application Menu
- Select the action you want to perform (e.g., create a definition)
- Select the resource type (e.g., create a DATABASE)
- Enter the data (e.g., enter the database name and attributes)

Using MR to Create Resources

- Multiple resources of the same type can be created with a single command
- Separate resource names by commas
- Two “views” available
 - List view: each parameter on a separate line with a short description
 - Command Syntax view: fewer screens and requires higher skill level

Using MR to Create Resources

- Scroll through the screens using ISPF function key assignments
 -  (scroll forward)
 -  (scroll back)
- Make changes to defaults on the screen and press Enter to submit
- A confirmation screen will be presented if elected in Preferences
- Press enter again and MR will submit command to IMS(s)
 - MR will edit input for valid field values
 - Does not cross check multiple fields
 - IMS edits input and returns success or failure
 - Now for an example of creating a database...

84

February 2009

If your preference is to confirm all updates, a confirmation screen will be displayed. Check the definitions and press Enter again. MR will submit a CREATE TRAN command for you.

Note that MR will edit fields for valid values but will not cross-check to be sure definitions are consistent – for example MR will not verify that a RTC is assigned to an FPE program, or that a FP transaction is not conversational. This is done only by IMS.

MR's Initial "Create New Resource" Panel

```
File Action Manage resources Spoc View Options Help
-----
Plex1                               IMS Create New Resource
Command ==>> _____
----- Plex. . ____ Route. . ____ Wait. . ____

Select a resource type. To base the resource on a template, specify
template information. Press Enter to continue.

* Resource. . . . . 1_ 1. Database
                   2. Program
                   3. Routing Code
                   4. Transaction

Resource name . . . . . CUSTMSTR _____
Resource type . . . . . 1_ 1. Resource
                   2. Descriptor

Definition template . . . . . 1 1. System default
                   2. Existing resource

Existing resource
  Definition template name . . . _____
  Definition template type . . . - 1. Resource
                                   2. Descriptor

F1=Help F3=Exit F4=Showlog F6=Expand F12=Cancel
```

Create database resource using system default as model.

Leave blank. Use current system default descriptor.

85

February 2009

Now lets create a database. Here we say we want to create a
-database
-resource
-using the current system transaction descriptor for fields
values we don't supply

MR's List View of Creating a Database

```
File Action Manage resources Spoc View Options Help
-----
Plex1                IMS Create Databases
Command ===> _____

----- Plex. . ____ Route. . ____ Wait. . ____
Press Enter to continue.

*NAME Database name. . . . .CUSTMSTR
ACCTYPE Access type . . . . .EXCL  EXCL, BRWS, READ, UPD
RESIDENT Resident in storage . N   Y, N

F1=Help  F3=Exit  F4=Showlog  F6=Expand  F12=Cancel
```

86

February 2009

The first set of screens uses the “List View”..Each parameter on a separate line with a short description

MR's Command Syntax View of Creating a Database

```
File Action Manage resources Spoc View Options Help
-----
Plex1                IMS Create Databases
Command ==> _____
----- Plex. . _____ Route. . _____ Wait. . _____

Press Enter to continue.

CREATE DB NAME ( CUSTMSTR )
SET ( ACCTYPE ( EXCL ) RESIDENT ( N ) )

F1=Help  F3=Exit  F4=Showlog  F6=Expand  F12=Cancel
```

87

February 2009

Command syntax view.

The second set of screens shows the same CREATE action using the “Command Syntax View”

- Fewer screens

- Requires higher skill level

Deleting Resources

- Same navigation as “create”
 - Select Action
 - Select Resource Type
 - Name resource(s) to be deleted
- Wild cards supported for delete processing
 - TEMP* - all resources beginning with TEMP
 - T%MP* - can also use single character substitution
 - * - all resources (be careful)
- Same panel format for deleting all resource and descriptor types
- Follow same rules for deleting as native DRD
 - Resource cannot be “in use”

88

February 2009

Deleting resources or descriptors follows the same general navigation path

- select an action (delete)
- select a type
- identify (name) the resource

For delete, wild cards are supported, including “*” which is delete ALL – be careful.

Updating Resources

- Update command
 - Supports changing resource or descriptor attributes
 - Does NOT support changing resource status (use SPOC)
- Same List or Command Syntax views available

89

February 2009

The MR application can be used to make updating definitions easier. You can't use MR to update status – only attributes.

Querying Resources

- Support for the QUERY command has been enhanced
 - Shortcut to displaying ALL attributes of a resource
 - QRY xxxx SHOW(ALL)
 - Shortcut to displaying resources with “exceptional” status
 - QRY xxxx <user-default filters>
 - Build customized queries
 - QRY xxxx <customized filters>
 - Field level help on all displayed fields
 - Put cursor on column heading and press F1

90

February 2009

MR also supports an enhanced QRY capability.

- a shortcut for displaying all attributes of a resource
- a shortcut for displaying resources with “exceptional” status (You get to decide what is exceptional)
- a screen for building customized queries
- an action column on the QRY results screen where you can enter line commands
- field level help for all displayed fields

MR Example of Querying a Transaction

```
File  Action  Manage resources  Spoc  View  Options  Help
-----
Plex1                                IMS Query Resources
Command ==> _____

----- Plex. . _____ Route. . _____ Wait. . _____

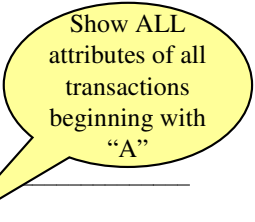
Select a resource type. Press Enter to continue.

* Resource. . . . . 4 1. Database
                  2. Program
                  3. Routing Code
                  4. Transaction

Resource name . . . . . A*
Resource type . . . . . 1 1. Resource
                  2. Descriptor

* Query type. . . . . 1 1. Show all
                  2. Exceptions
                  3. Custom

F1=Help  F3=Exit  F4=Showlog  F6=Expand  F12=Cancel
```



91

February 2009

- From this screen you can enter
- resource type (transaction)
 - resource name or names (A*)
 - is it a resource or descriptor
 - short cut – SHOW(ALL)

MR Example of Command Response for Query

```

File  Action  Manage resources  Spoc  View  Options  Help
-----
Plex1                                IMS Single Point of Control
Command ==>> _____

----- Plex. . _____ Route. . _____ Wait. . _____

Response for:  QRY TRAN NAME(A*) SHOW ALL                               More:  >
Trancode MbrName  CC PSBname  LCLs  LQCnt  LLCT  LPLCT  LPLCTTime  LCPRI
ADDPART  SYS3      0 DFSSAM04   4      0      2  65535    65535      7
AOBMP    SYS3      0 TS2IAOB0  23      0  65535  65535    65535      7
AOP      SYS3      0 TS1IAOP0   4      0      4    4      5      0
APOL11   SYS3      0 APOL1      1      0  65535  65535    65535     10
APOL13   SYS3      0 APOL1      1      0  65535  65535    65535      1
APOL14   SYS3      0 APOL1      1      0  65535  65535    65535      9
APOL15   SYS3      0 APOL1      1      0  65535  65535    65535      1
APOL16   SYS3      0 APOL1      1      0  65535  65535    65535      1
APOL17   SYS3      0 APOL1      1      0  65535  65535    65535      1
APOL18   SYS3      0 APOL1      1      0  65535  65535    65535      1

F1=Help  F3=Exit  F4=Showlog  F6=Expand  F12=Cancel

```

92

February 2009

The QRY results screen has a new Action column in which you can put line commands. Line commands are shown near the top of the screen as C, D, U, and V. Each line command will be executed serially – that is, MR will submit one, then the next, etc.

If you put a “/” in the action column, you can then pull down the Action Menu from the Action Bar at the top and select an action. That action will be executed for every resource selected.

Disclaimer

*© Copyright IBM Corporation 2009. All rights reserved.
U.S. Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule
Contract with IBM Corp.*

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS AND/OR SOFTWARE.

IBM, the IBM logo, ibm.com, and IMS are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml

93

February 2009