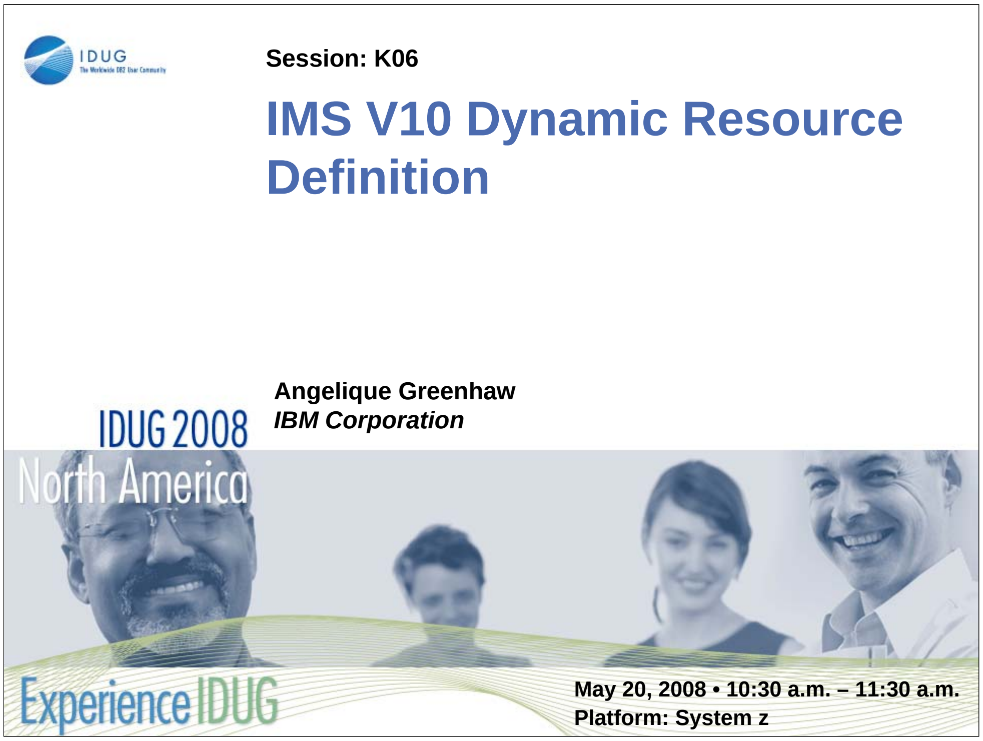


Session: K06

IMS V10 Dynamic Resource Definition

Angelique Greenhaw
IBM Corporation

May 20, 2008 • 10:30 a.m. – 11:30 a.m.
Platform: System z

IMS™ Version 10, announced in October 2006, focuses on improved integration, manageability, and scalability for this premier database and transaction management system. It offers enhancements to all major components to help ensure the growth, availability and systems management that is needed for on-demand business environments. This session describes Dynamic Resource Definition (DRD), a major enhancement that allows for dynamic definition of MODBLKS resources (application programs, transactions, databases, Fast Path routing codes) to help improve IMS availability. This removes the requirement for a MODBLKS SYSGEN and MODBLKS online change.

Overview

- Dynamic Resource Definition (DRD)
  - – MODBLKS resource definitions
  - – Modifying resource definitions without DRD
  - – DRD objective
  - – DRD components
  - – Modifying resource definitions with DRD
  - – MODBLKS resource loading during IMS restart
  - – DRD environment
  - – Enabling DRD
  - – New V10 terminology for MODBLKS resources
  - – MODBLKS and RDDS data sets
  - – RDDS Extraction utility

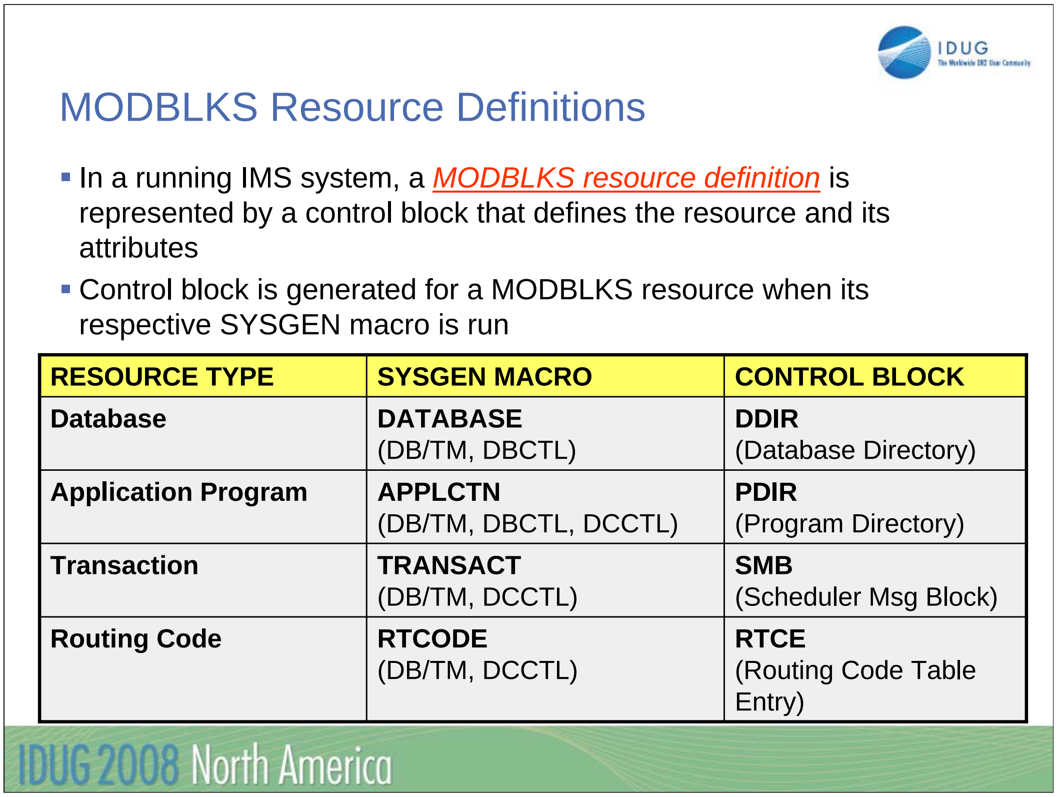
Overview (continued)

- Dynamic Resource Definition (DRD)
  - Descriptors
  - Recoverability of resources/descriptors
  - Commands/How to use – CREATE, DELETE, UPDATE, QUERY

- Manage Resources (MR) application
  - Guides user through DRD process
  - Example screenshots

- Summary of Dynamic Resource Definition and Manage Resources

Dynamic Resource Definition

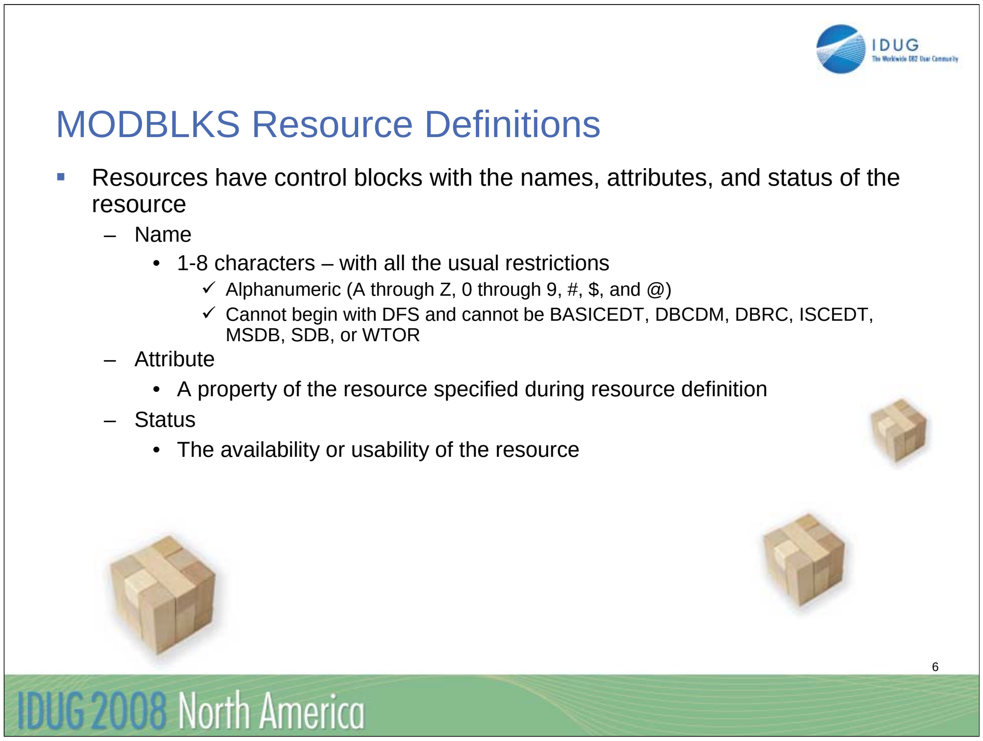


MODBLKS Resource Definitions

- In a running IMS system, a *MODBLKS resource definition* is represented by a control block that defines the resource and its attributes
- Control block is generated for a MODBLKS resource when its respective SYSGEN macro is run

RESOURCE TYPE	SYSGEN MACRO	CONTROL BLOCK
Database	**DATABASE** (DB/TM, DBCTL)	**DDIR** (Database Directory)
Application Program	**APPLCTN** (DB/TM, DBCTL, DCCTL)	**PDIR** (Program Directory)
Transaction	**TRANSACT** (DB/TM, DCCTL)	**SMB** (Scheduler Msg Block)
Routing Code	**RTCODE** (DB/TM, DCCTL)	**RTCE** (Routing Code Table Entry)

What do we mean by “MODBLKS resource definitions”? A MODBLKS resource can be either a database, program (PSB/Application), transaction, and/or routing code. They are represented by control blocks in the IMS control region, or a running IMS system. This concept has not changed for DRD. The table showed on this slide displays the different MODBLKS resource types, along with their respective SYSGEN macro, and the control block that is generated for that resource when its respective SYSGEN macro is run.



MODBLKS Resource Definitions

- Resources have control blocks with the names, attributes, and status of the resource
  - Name
    - 1-8 characters – with all the usual restrictions
      - ✓ Alphanumeric (A through Z, 0 through 9, #, $, and @)
      - ✓ Cannot begin with DFS and cannot be BASICEDT, DBCDM, DBRC, ISCEDT, MSDB, SDB, or WTOR
  - Attribute
    - A property of the resource specified during resource definition
  - Status
    - The availability or usability of the resource

6

IDUG 2008 North America

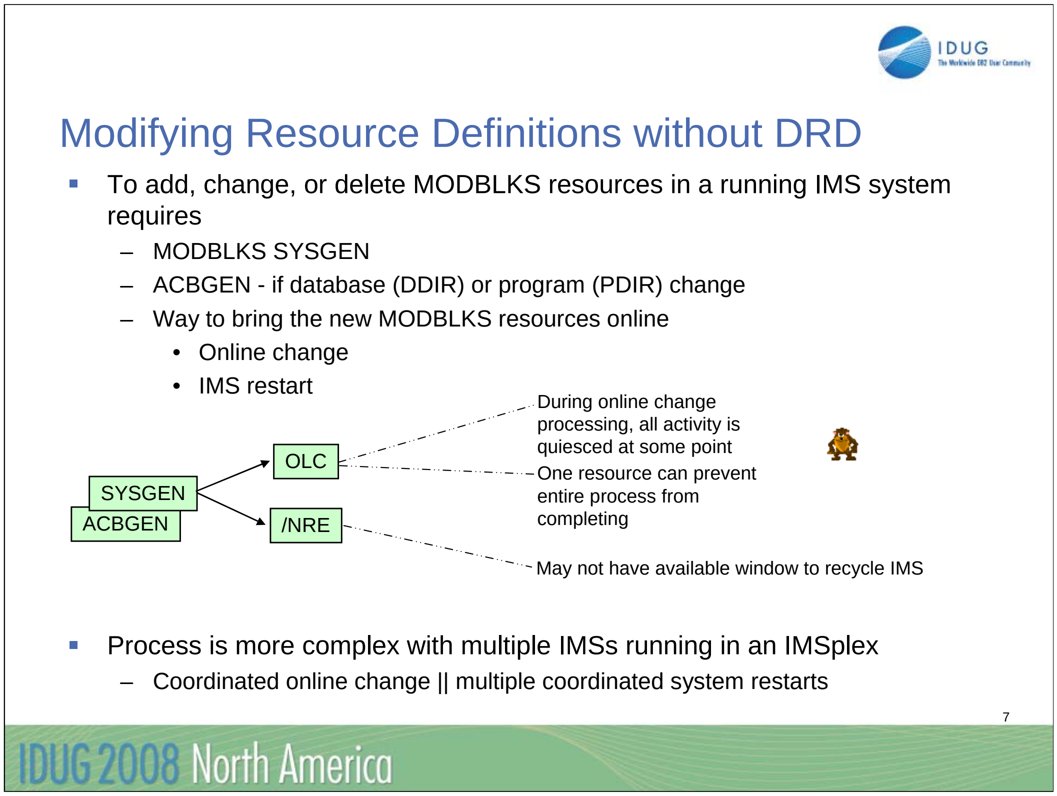
Resources themselves have names, attributes, and status. There is no change to the naming conventions or requirements with DRD. An attribute is a property of a resource. For example, a transaction has an attribute – MAXRGN – defining how many regions that transaction can be scheduled into. Another attribute is whether that transaction is conversational. Resources also have statuses. There are several different statuses for each resource, but generally they indicate whether or not the resource is available for use.

More detailed elaboration:

This slide talks about attributes – or “properties” of a resources. Examples are the ACCESS attribute for a DATABASE, which in DRD is referred to as ACCTYPE (access type). RESIDENT is another database attribute.

The resource definition process also has default attributes which are used when the user does not completely define all attributes in the sysgen macro or the CREATE command. For example, if you do not include ACCESS in the DATABASE macro, it defaults to EX (exclusive). If you do not SET the ACCTYPE attribute in the CREATE command, it will default to whatever is in the system ”default descriptor” – discussed later. Note that users can set the default attributes to any valid value – they are not bound by what IMS thinks the defaults should be.

Status describes a property of a resource which generally define it availability



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 restart

SYSGEN
ACBGEN
OLC
/NRE

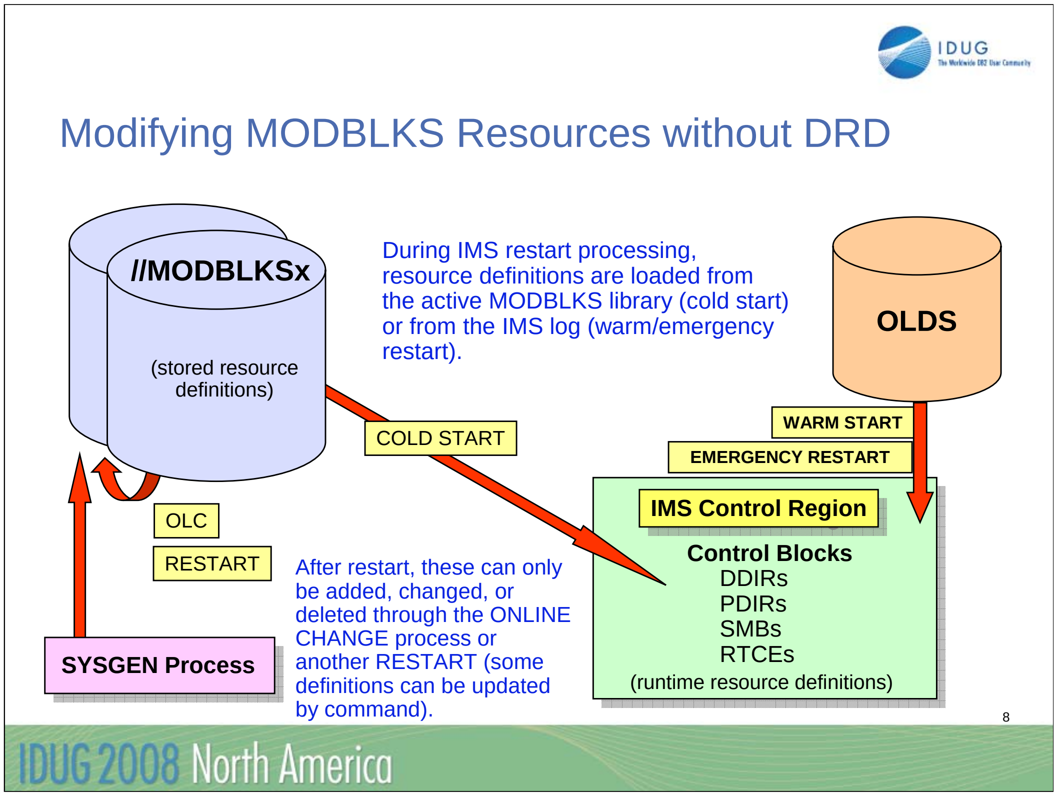
During online change processing, all activity is quiesced at some point

One resource can prevent entire process from completing

May not have available window to recycle IMS

- Process is more complex with multiple IMSs running in an IMSplex
  - Coordinated online change || multiple coordinated system restarts

Prior to DRD, to add, change, or delete a resource required an IMS sysgen and either an IMS restart or an online change. Both of these processes resulted in significant (or at least noticeable) unavailability to the end user. In a multi-IMS plex, it was even more complex because of the need to coordinated the changes (and the unavailability) across multiple IMSs.



Modifying MODBLKS Resources without DRD

//MODBLKSx

(stored resource definitions)

During IMS restart processing, resource definitions are loaded from the active MODBLKS library (cold start) or from the IMS log (warm/emergency restart).

OLDS

COLD START

WARM START

EMERGENCY RESTART

IMS Control Region

Control Blocks
DDIRs
PDIRs
SMBs
RTCEs
(runtime resource definitions)

OLC

RESTART

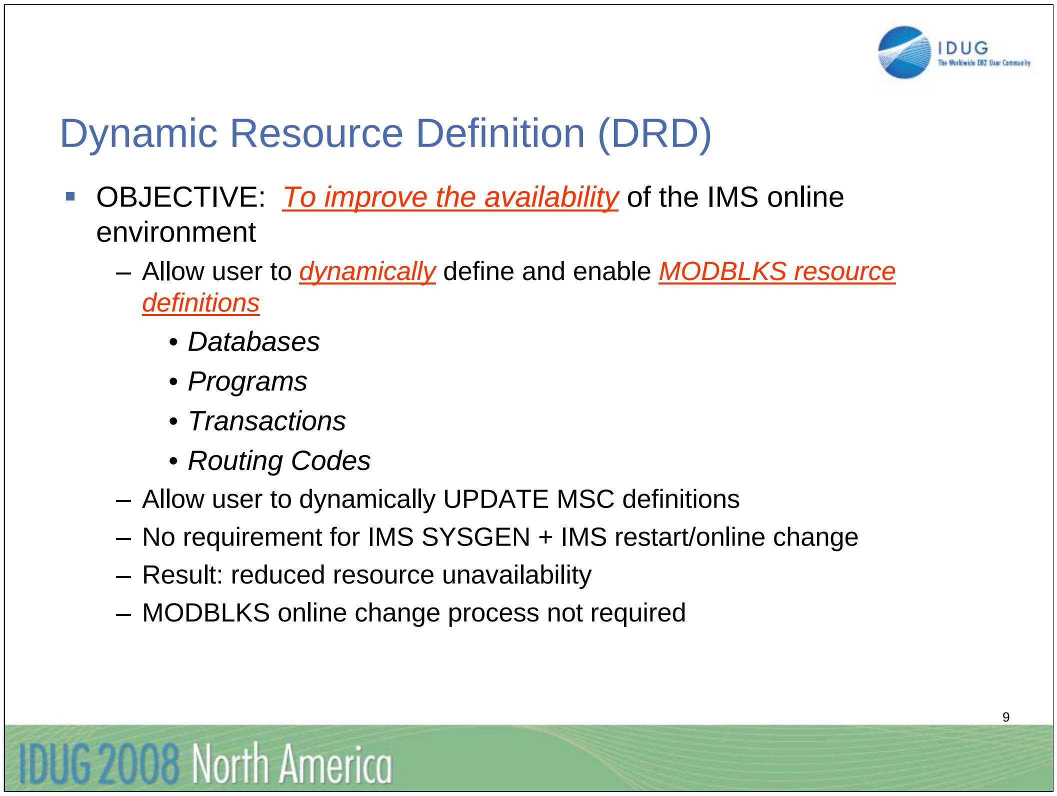
SYSGEN Process

After restart, these can only be added, changed, or deleted through the ONLINE CHANGE process or another RESTART (some definitions can be updated by command).

Without DRD, a MODBLKS online change, or a sysgen and 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 these point, these definitions, or control blocks, can only be modified by doing another SYSGEN plus an online change or IMS restart. And I just mentioned 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).

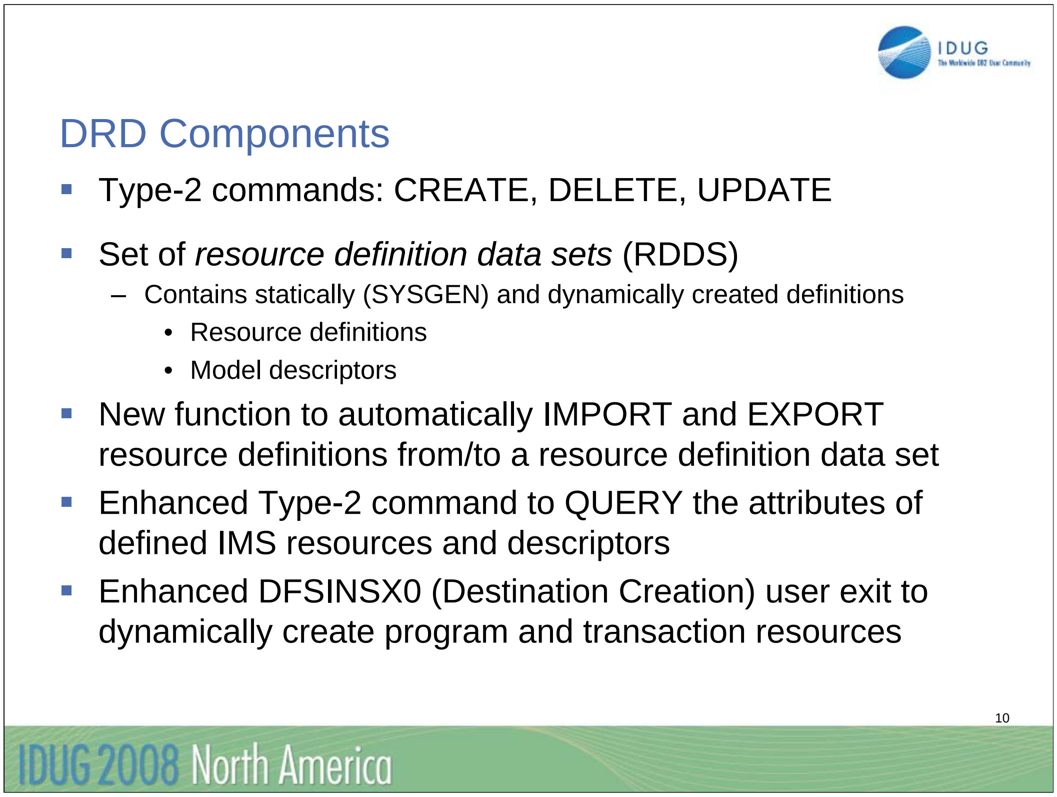


Dynamic Resource Definition (DRD)

- OBJECTIVE: *To improve the availability* of the IMS online environment
  - Allow user to *dynamically* define and enable *MODBLKS resource definitions*
    - *Databases*
    - *Programs*
    - *Transactions*
    - *Routing Codes*
  - Allow user to dynamically UPDATE MSC definitions
  - No requirement for IMS SYSGEN + IMS restart/online change
  - Result: reduced resource unavailability
  - MODBLKS online change process not required

The object, therefore, of Dynamic Resource Definition, is to improve the availability of IMS by allowing MODBLKS resource changes without a sysgen or OLC. The way it does this is by allowing the user to dynamically create and enable MODBLKS resources, which are listed here. These are your databases, programs, transactions and routing codes. {{DDIR is the Database Directory, PDIR is the Program Directory, SMB is the Scheduler Message Block, and RTCE is the Routing Code Table Entry (for fast path EMH routing codes).}}

Another part of DRD is that you can dynamically update your MSC definitions. When you use DRD, there is no need to run an IMS SYSGEN + the online change or restart. The availability issues that arise from those methods of managing your MODBLKS resources are eliminated. And of course, you no longer need to use MOBDLKS online change.



DRD Components

- Type-2 commands: CREATE, DELETE, UPDATE
- Set of *resource definition data sets* (RDDS)
  - Contains statically (SYSGEN) and dynamically created definitions
    - Resource definitions
    - Model descriptors
- New function to automatically IMPORT and EXPORT resource definitions from/to a resource definition data set
- Enhanced Type-2 command to QUERY the attributes of defined IMS resources and descriptors
- Enhanced DFSINSX0 (Destination Creation) user exit to dynamically create program and transaction resources

The major components of DRD include:

-one new (CREATE) and several commands (UPDATE, DELETE) which are not new but have new functionality.

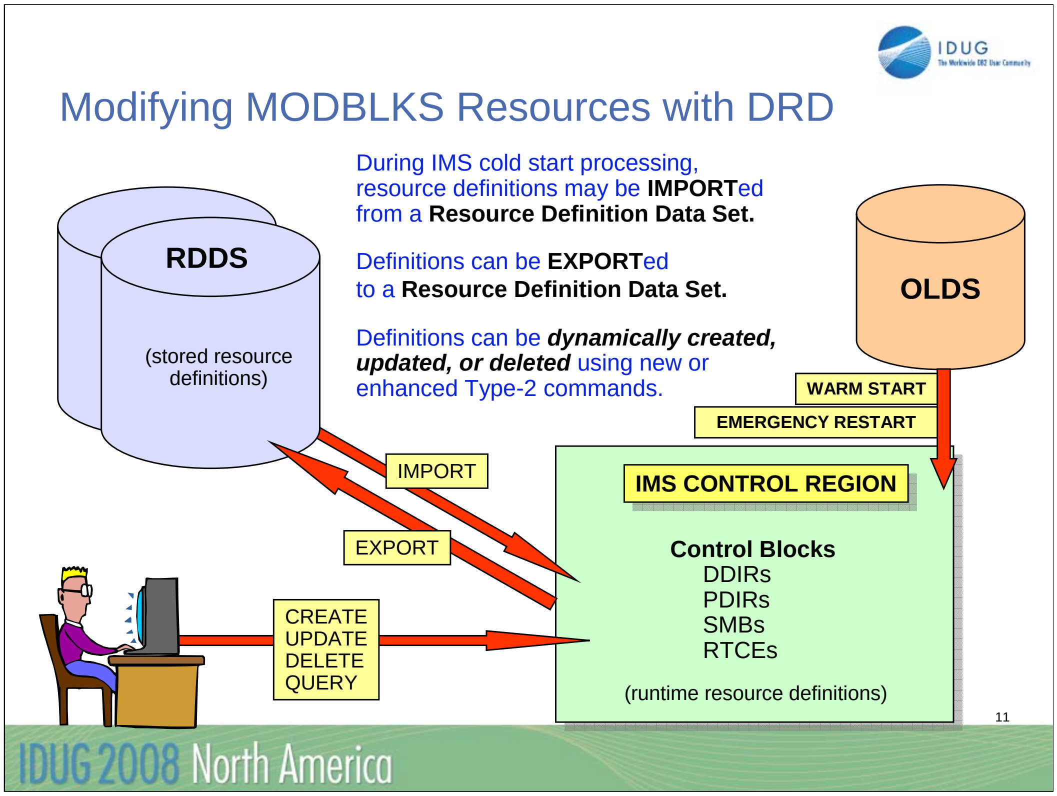
-a set of BSAM data sets used to contain resource definitions from which IMS may import or to which IMS may export

-an automatic import function to enable IMS to determine the source of its definitions at cold start

-an automatic export function to enable IMS to save new and changed definitions to the RDDS to that the next cold start can import them

-enhancements to the QRY command to show ALL resource and descriptor attributes and 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 (addressed in Sys Mgmt) is entered with an unknown transaction code



Modifying MODBLKS Resources with DRD

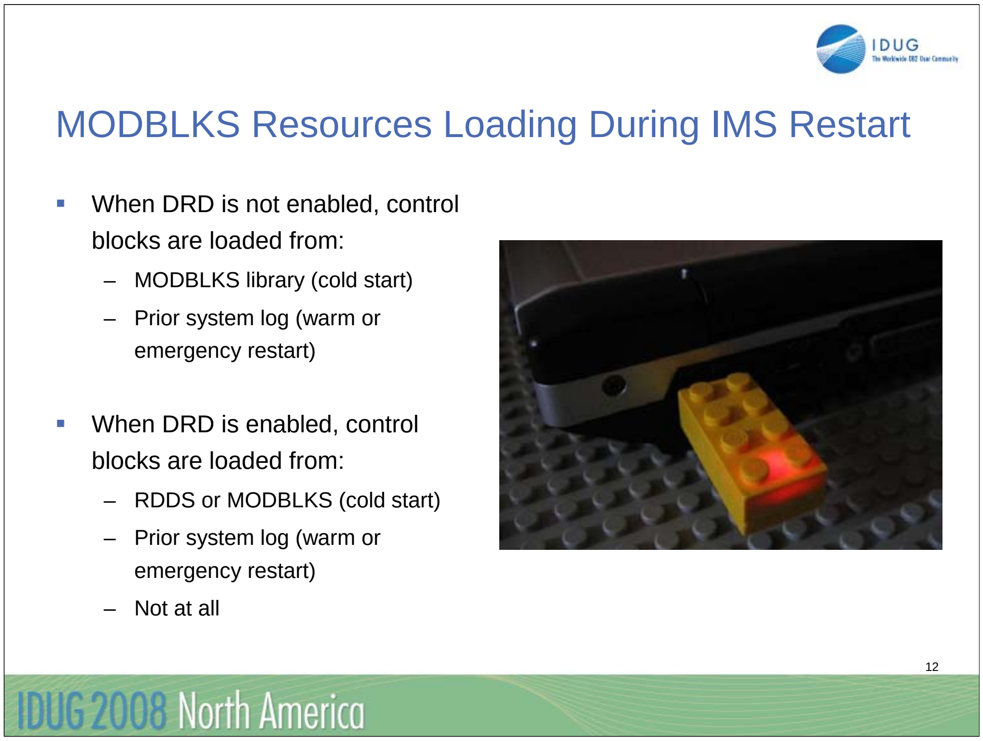
During IMS cold start processing, resource definitions may be **IMPORT**ed from a **Resource Definition Data Set.**

Definitions can be **EXPORT**ed to a **Resource Definition Data Set.**

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

This chart shows the SYSGEN process being replaced by a user entering Type-2 commands, and the cold start being replaced by an IMPORT function. EXPORT allows resources defined by the user dynamically to be saved in an RDDS for the next IMS cold start.

MODBLKS resources will be EXPORTed to the RDDS at every system checkpoint. They will be written to the IMS log as they are created, deleted or updated by the DRD process…or any time there is a system checkpoint.

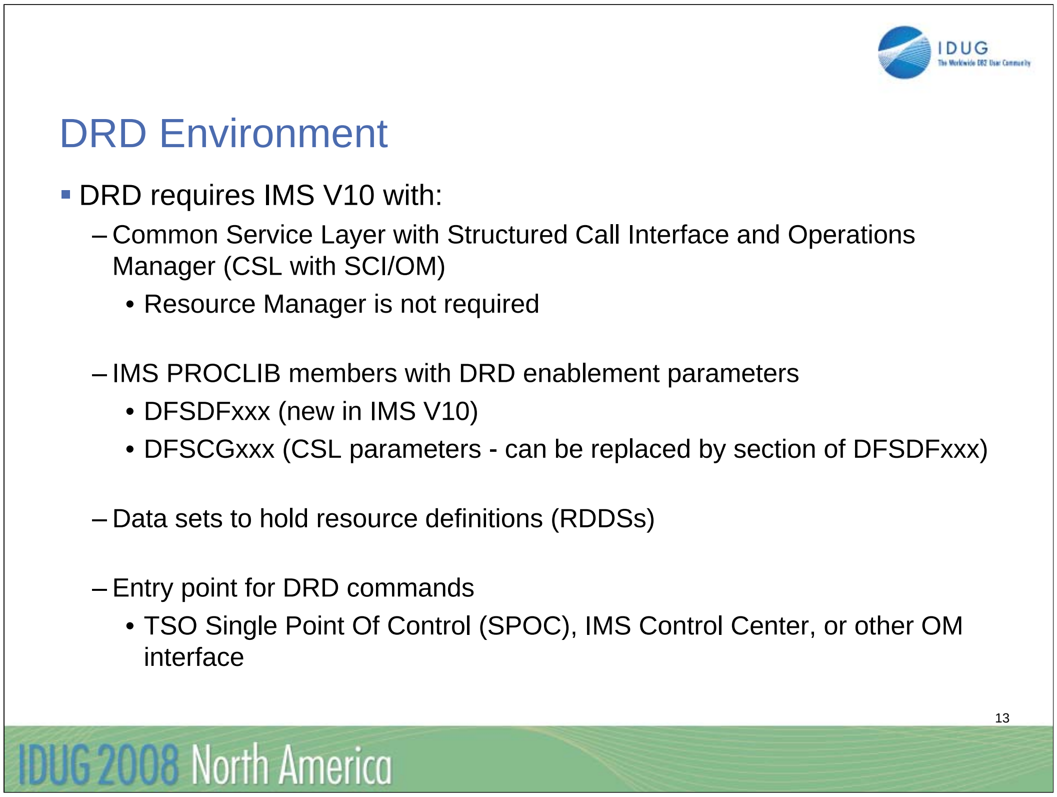


MODBLKS Resources Loading During IMS Restart

- When DRD is not enabled, control blocks are loaded from:
  - MODBLKS library (cold start)
  - Prior system log (warm or emergency restart)
- When DRD is enabled, control blocks are loaded from:
  - RDDS or MODBLKS (cold start)
  - Prior system log (warm or emergency restart)
  - Not at all

When DRD is not enabled, the control blocks are loaded from MODBLKS during IMS cold start. When DRD is not enabled, they may be loaded from an RDDS or from MODBLKS (or not at all). During warm or emergency restart, control blocks are loaded from the IMS logs regardless of whether DRD is enabled or not.

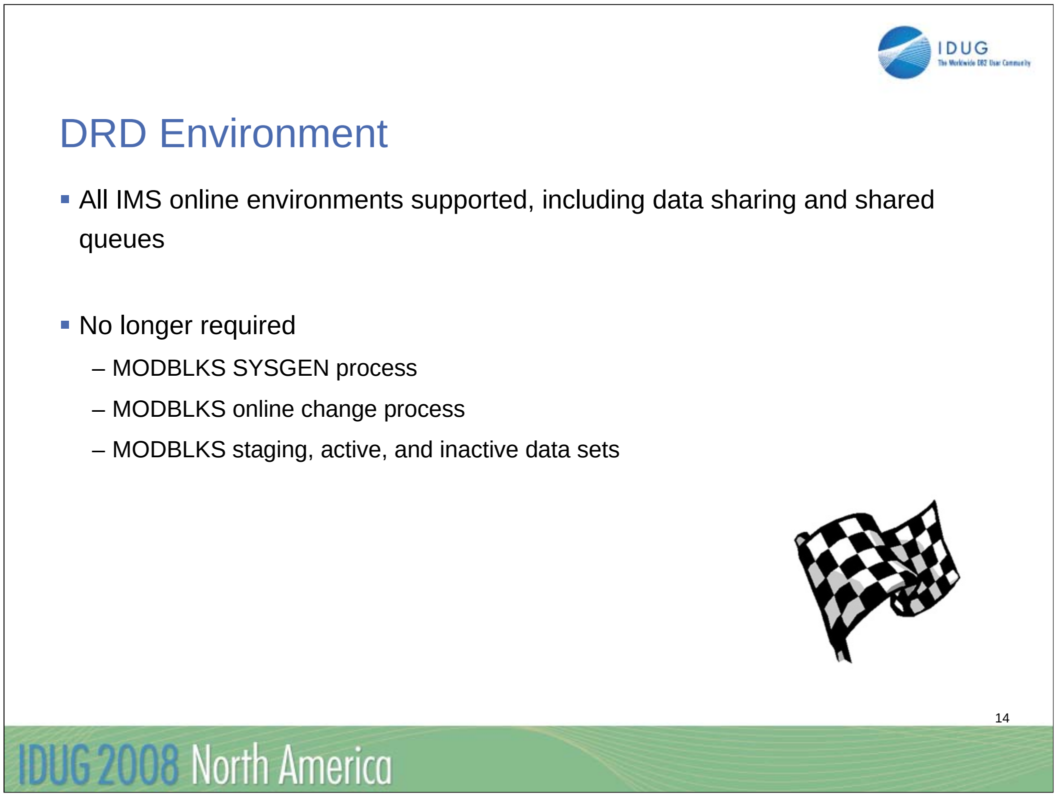
During execution, resources may be created, deleted, or updated online. These changes are logged in x’22’ log records. At system checkpoint, including initialization and shut down checkpoints, all resources are logged and optionally written to the RDDS (depending on the options set for AUTOEXPORT in DFSDFxxx).



DRD Environment

- DRD requires IMS V10 with:
  - Common Service Layer with Structured Call Interface and Operations Manager (CSL with SCI/OM)
    - Resource Manager is not required
  - IMS PROCLIB members with DRD enablement parameters
    - DFSDFxxx (new in IMS V10)
    - DFSCGxxx (CSL parameters - can be replaced by section of DFSDFxxx)
  - Data sets to hold resource definitions (RDDSs)
  - Entry point for DRD commands
    - TSO Single Point Of Control (SPOC), IMS Control Center, or other OM interface

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 and entry point for the DRD commands (probably the SPOC or Manage Resources application)

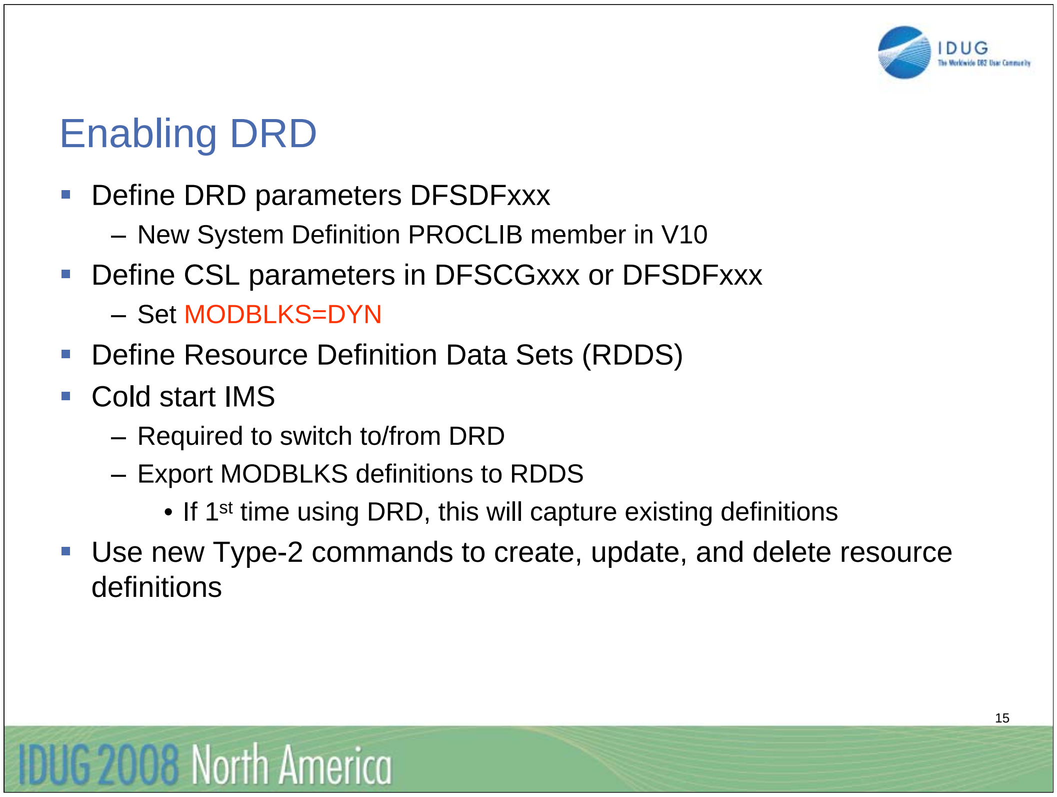


DRD Environment

- All IMS online environments supported, including data sharing and shared queues
- No longer required
  - MODBLKS SYSGEN process
  - MODBLKS online change process
  - MODBLKS staging, active, and inactive data sets

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

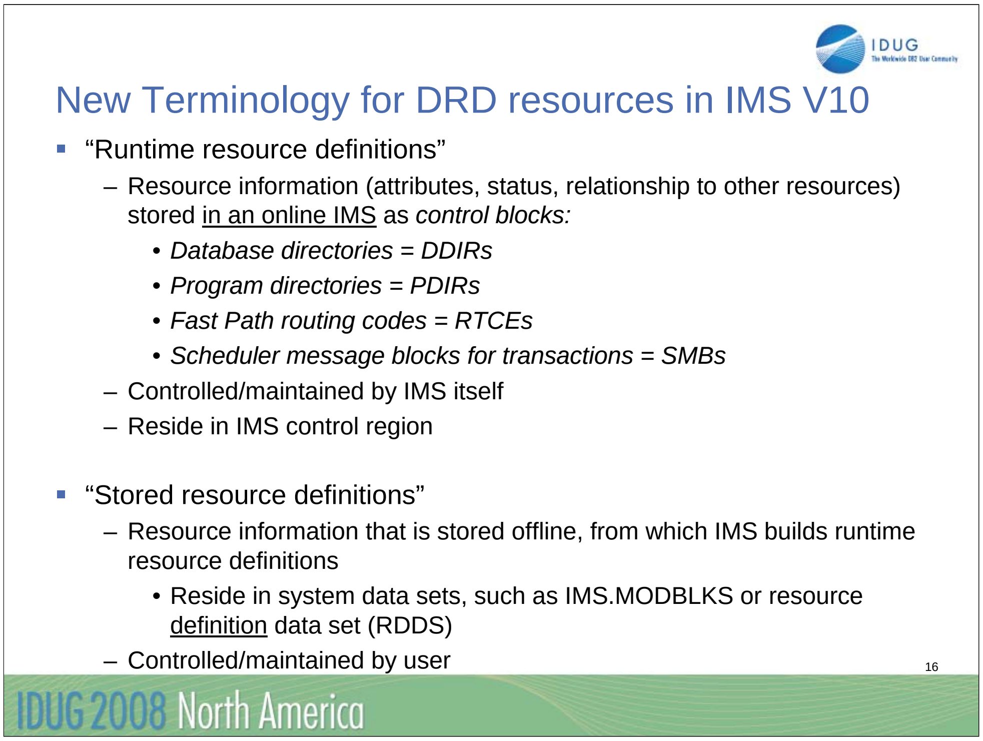
When DRD is fully implemented, there is no longer a need for a MODBLKS sysgen, the MODBLKS libraries associated with the sysgen or OLC process, nor is there a need for the MODBLKS DD statements in the IMS online JCL.



Enabling DRD

- Define DRD parameters DFSDFxxx
  - New System Definition PROCLIB member in V10
- Define CSL parameters in DFSCGxxx or DFSDFxxx
  - Set MODBLKS=DYN
- Define Resource Definition Data Sets (RDDS)
- Cold start IMS
  - Required to switch to/from DRD
  - Export MODBLKS definitions to RDDS
    - If 1st time using DRD, this will capture existing definitions
- Use new Type-2 commands to create, update, and delete resource definitions

There is a new proclib member in V10 called the System Definition member. This member contains the DRD parameters (for example, AUTOIMPORT and AUTOEXPORT options and RDDS data sets names). DRFD is enabled by saying that MODBLKS blocks are Dynamic. This is done in the CSL definitions (the CSL library can be replaced by a CSL section in DFSDFxxx). After defining the RDDSs (and cataloging – they are dynamically allocated by the control region), a cold start is required the first time you come up with DRD. During the cold start, IMS can load existing definitions from MODBLKS and then automatically EXPORT them to an RDDS. The RDDS can then be used in future cold start to load these definitions.



IDUG The Worldwide DB2 User Community

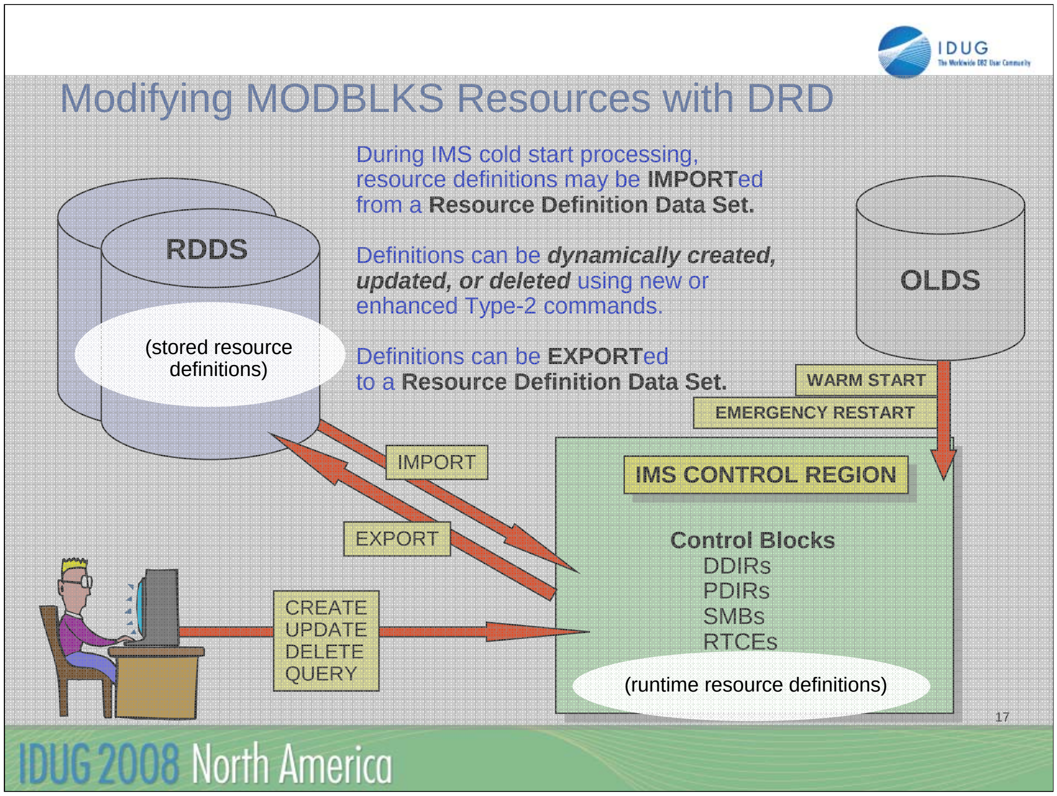
New Terminology for DRD resources in IMS V10

- “Runtime resource definitions”
  - Resource information (attributes, status, relationship to other resources) stored in an online IMS as *control blocks:*
    - *Database directories = DDIRs*
    - *Program directories = PDIRs*
    - *Fast Path routing codes = RTCEs*
    - *Scheduler message blocks for transactions = SMBs*
  - Controlled/maintained by IMS itself
  - Reside in IMS control region
- “Stored resource definitions”
  - Resource information that is stored offline, from which IMS builds runtime resource definitions
    - Reside in system data sets, such as IMS.MODBLKS or resource definition data set (RDDS)
  - Controlled/maintained by user

In V10, we are starting to use some new terminology when we were to MODBLKS resources. The new terminology is more granular in that it now relates to where the MODBLKS resources exist – either online or offline.

“Runtime resource definitions” are those MODBLKS resources that exist in a running IMS system as control blocks. They are controlled by IMS and reside in the IMS control region.

On the other hand, “stored resource definitions” are those MODBLKS resources that are stored offline in a dataset, such as the MODBLKS dataset or the RDDS if you’re using DRD. These definitions are read by IMS either during coldstart or automatic import and IMS uses them to build the runtime resource definitons, which I just mentioned were stored in the IMS control region as control blocks. Stored resource definitions are controlled and maintained by the user.



Modifying MODBLKS Resources with DRD

During IMS cold start processing, resource definitions may be **IMPORT**ed from a **Resource Definition Data Set.**

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

Definitions can be **EXPORT**ed to a **Resource Definition Data Set.**

RDDS

(stored resource definitions)

OLDS

WARM START

EMERGENCY RESTART

IMPORT

EXPORT

CREATE
UPDATE
DELETE
QUERY

IMS CONTROL REGION

Control Blocks
DDIRs
PDIRs
SMBs
RTCEs

(runtime resource definitions)

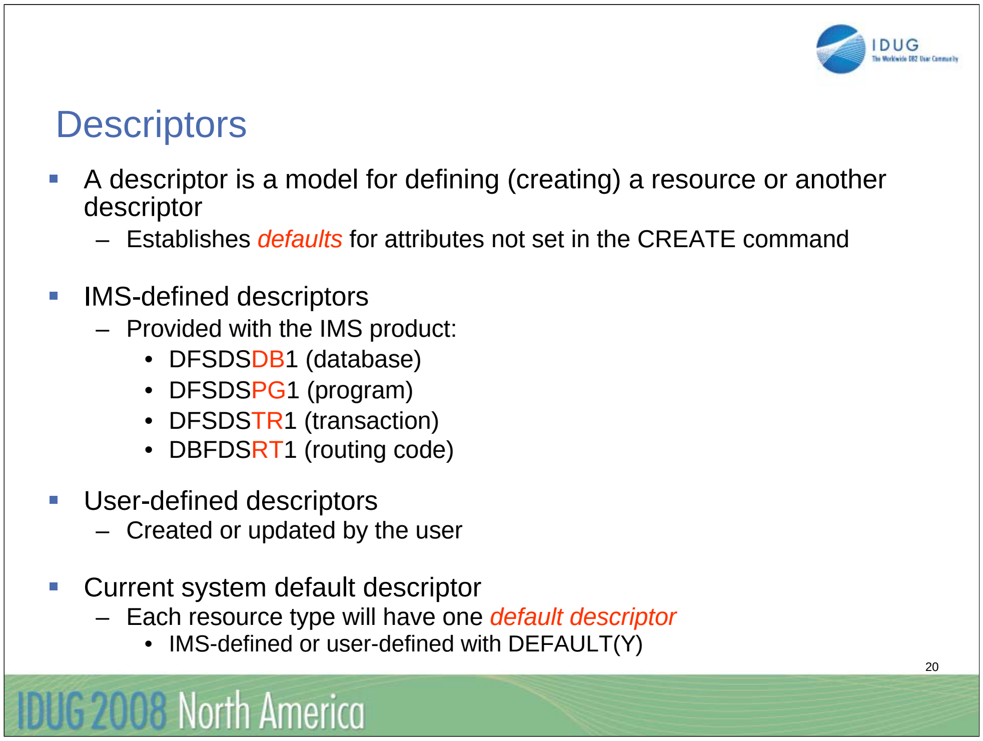
Here is how the new V10 terminology fits into the diagram that I showed earlier, depicting how to manage your MODBLKS resources using the DRD capability.

MODBLKS and RDDS Data Sets

- MODBLKS data set (typically used for online change -- no DRD)
  - Contains resource information from SYSGEN macros
    - DATABASE, APPLCTN, TRANSACT, RTCODE
- RDDS data set (used with DRD)
  - Contains resource information from SYSGEN macros
    - DATABASE, APPLCTN, TRANSACT, RTCODE
  - Contains resource information from dynamically created resources
    - Using command CREATE DB | PGM | TRAN | RTC
- Stored resource definition information format is different in MODBLKS vs. RDDS
- New utility for converting RDDS information to SYSGEN or type-2 command format

RDDS Extraction Utility (DFSURDD0)

- Offline batch utility
- Converts stored resource definitions in an RDDS to
  - IMS Stage-1 macro statements (APPLCTN, TRANSACT, DATABASE, RTCODE)
  - IMS Type-2 CREATE commands for DRD:
    - CREATE DB
    - CREATE TRAN
    - CREATE PGM
    - CREATE RTC
- Output written to SYSPRINT data set
- CREATE commands could be submitted to IMS via the batch SPOC
- Focus is to ease migration to/from DRD during DRD implementation and provide interface to pre-V10 existing procedures



Descriptors

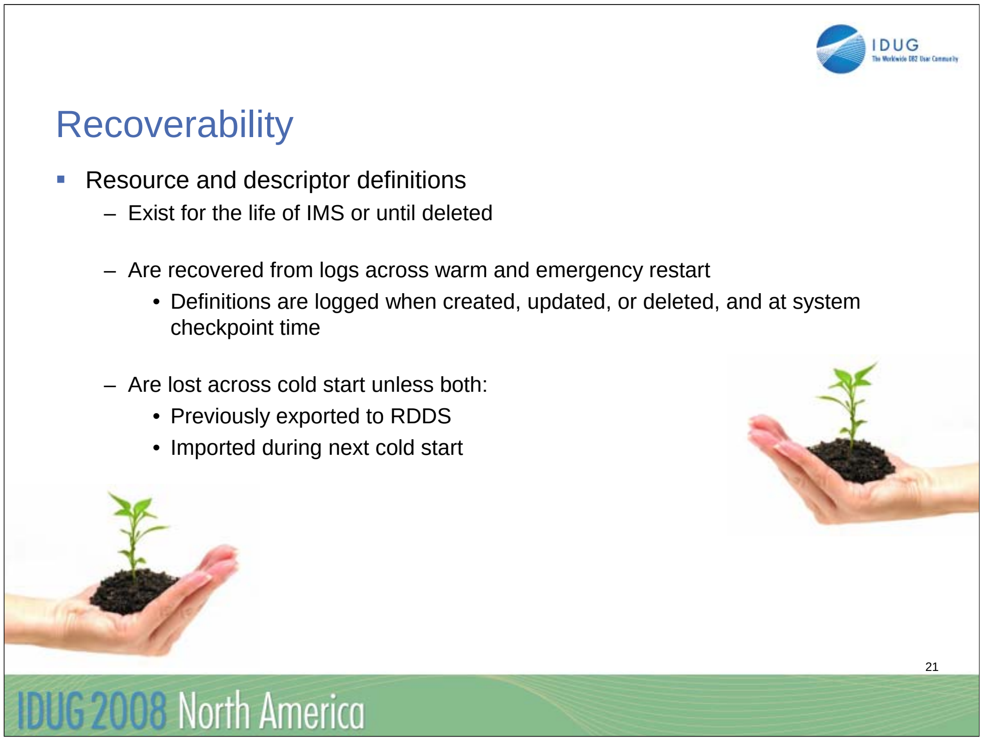
- A descriptor is a model for defining (creating) a resource or another descriptor
  - Establishes *defaults* for attributes not set in the CREATE command
- IMS-defined descriptors
  - Provided with the IMS product:
    - DFSDSDB1 (database)
    - DFSDSPG1 (program)
    - DFSDSTR1 (transaction)
    - DBFDSRT1 (routing code)
- User-defined descriptors
  - Created or updated by the user
- Current system default descriptor
  - Each resource type will have one *default descriptor*
    - IMS-defined or user-defined with DEFAULT(Y)

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 system 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 attributes in the sysgen macro or the CREATE command. For example, if you do not include ACCESS in the DATABASE macro, it defaults to EX (exclusive). If you do not SET the ACCTYPE attribute in the CREATE command, it will default to whatever is in the system "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

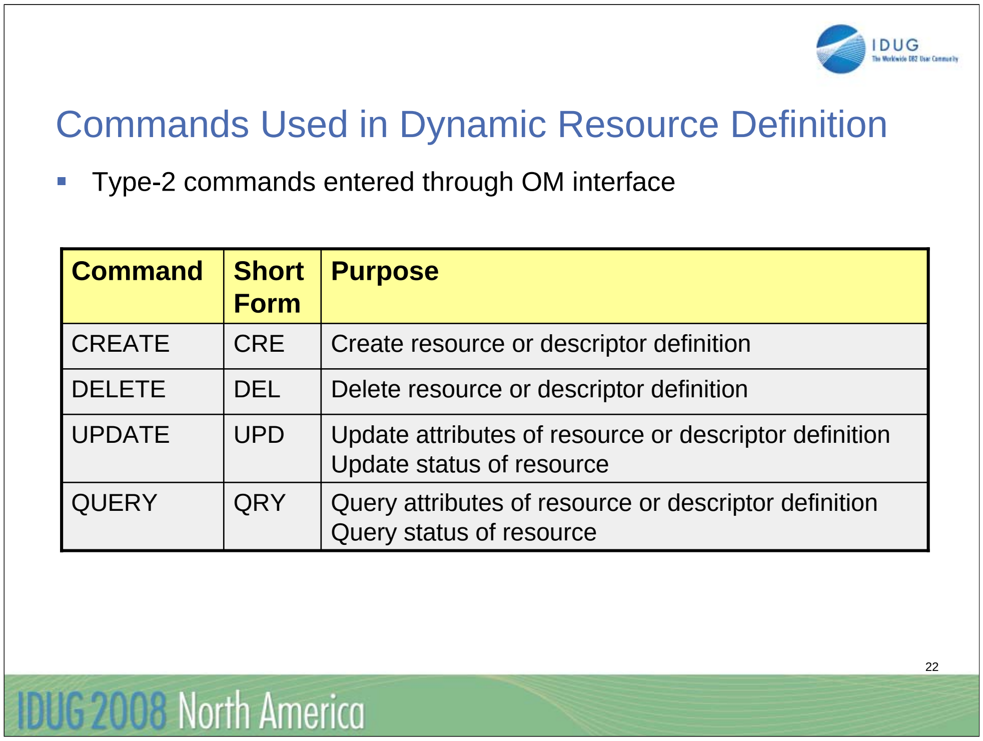


Recoverability

- 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 lost across cold start unless both:
    - Previously exported to RDDS
    - Imported during next cold start

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.



Commands Used in Dynamic Resource Definition

- Type-2 commands entered through OM interface

Command	Short Form	Purpose
CREATE	CRE	Create resource or descriptor definition
DELETE	DEL	Delete resource or descriptor definition
UPDATE	UPD	Update attributes of resource or descriptor definition Update status of resource
QUERY	QRY	Query attributes of resource or descriptor definition Query status of resource

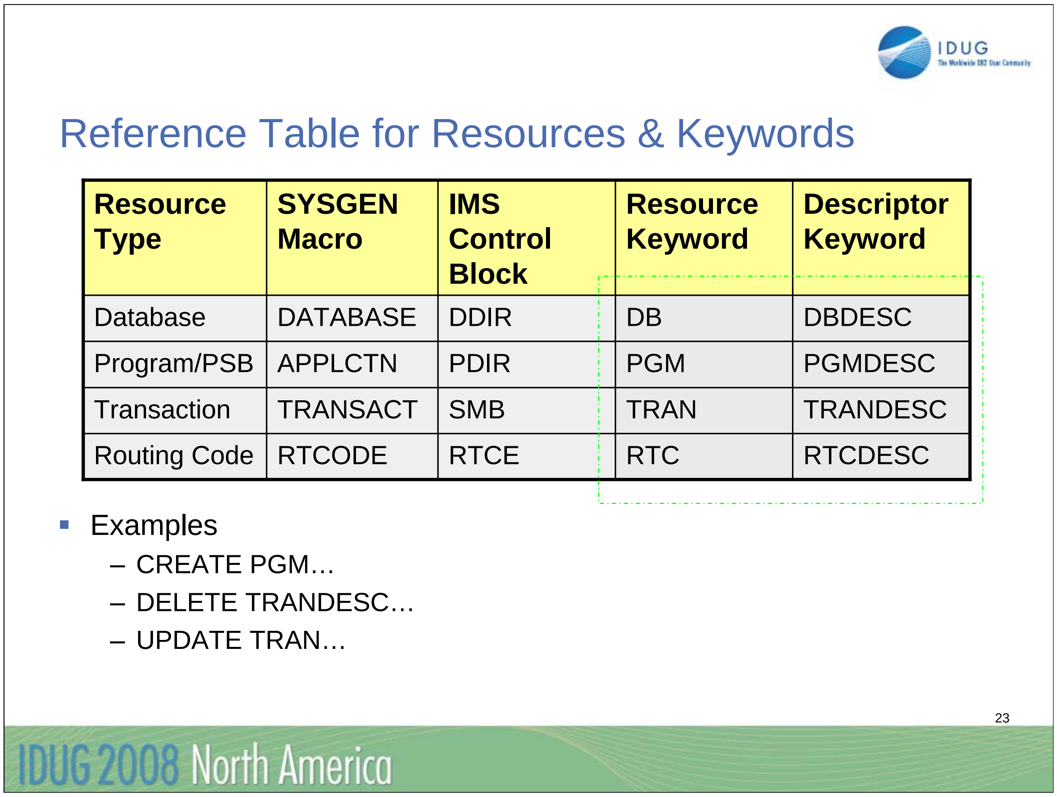
There are four main commands that are part of DRD.

-CREATE – this is the only new command available immediately. It is used to create new resources or descriptors

-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. The details of the UPDATE command will be presented mostly in the Systems Management part of this class

-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.”



Reference Table for Resources & Keywords

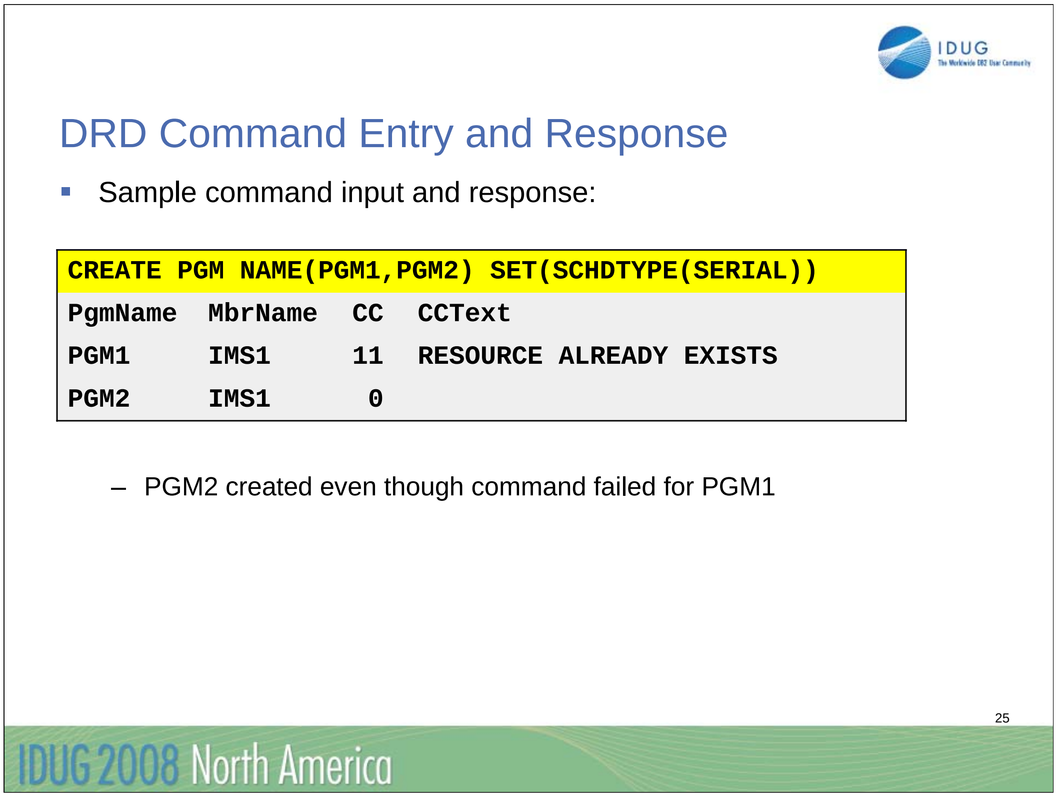
Resource Type	SYSGEN Macro	IMS Control Block	Resource Keyword	Descriptor Keyword
Database	DATABASE	DDIR	DB	DBDESC
Program/PSB	APPLCTN	PDIR	PGM	PGMDESC
Transaction	TRANSACT	SMB	TRAN	TRANDESC
Routing Code	RTCODE	RTCE	RTC	RTCDESC

- Examples
  - CREATE PGM…
  - DELETE TRANDESC…
  - UPDATE TRAN…

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



DRD Command Entry and Response

- Sample command input and response:

```
CREATE PGM NAME(PGM1,PGM2) SET(SCHDTYPE(SERIAL))
PgmName  MbrName  CC  CCText
PGM1     IMS1     11  RESOURCE ALREADY EXISTS
PGM2     IMS1      0
```

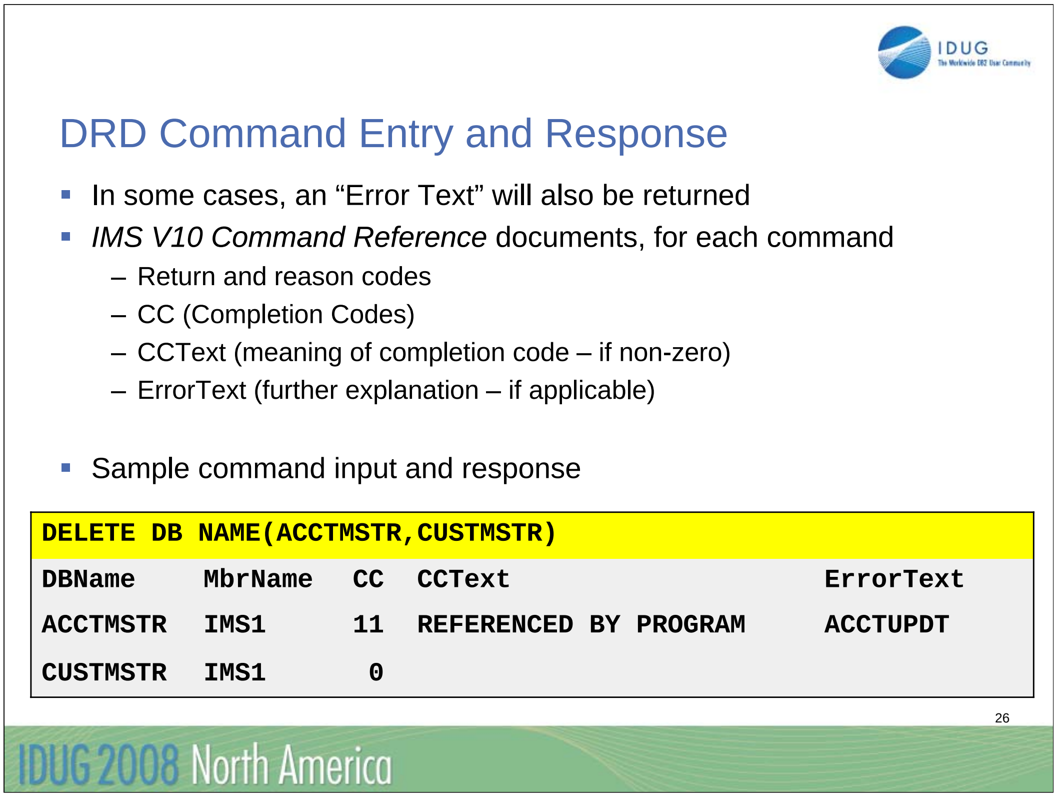
  – PGM2 created even though command failed for PGM1

The base format for all these commands is shown here. Can enter more than one NAME in the command, separated by commas. Except for CREATE, wild cards can be used, including an “*” meaning ALL of that type (be careful). The action applies to each resource NAMEd.

These are Type-2 commands and can only be entered through the OM interface, meaning IMS must be running in a CSL environment with a minimum of SCI and OM. As with other commands, OM will route the commands as requested, then wait a prescribed amount of time for the target IMSs to respond. It will then consolidate the responses and return to the entry point (e.g., the SPOC).

If multiple resources (or descriptors) are being created, updated, or deleted in the same command, it is possible that some might succeed while others fail. Each resource is acted upon independently. You may get a response indicating that (for example) creation of PGM1 failed because PGM1 already exists, but creation of PGM2 was successful.

Like are responses through OM, IMS returns the response encapsulated in XML tags. The example shown on this slide has been formatted by the SPOC application.



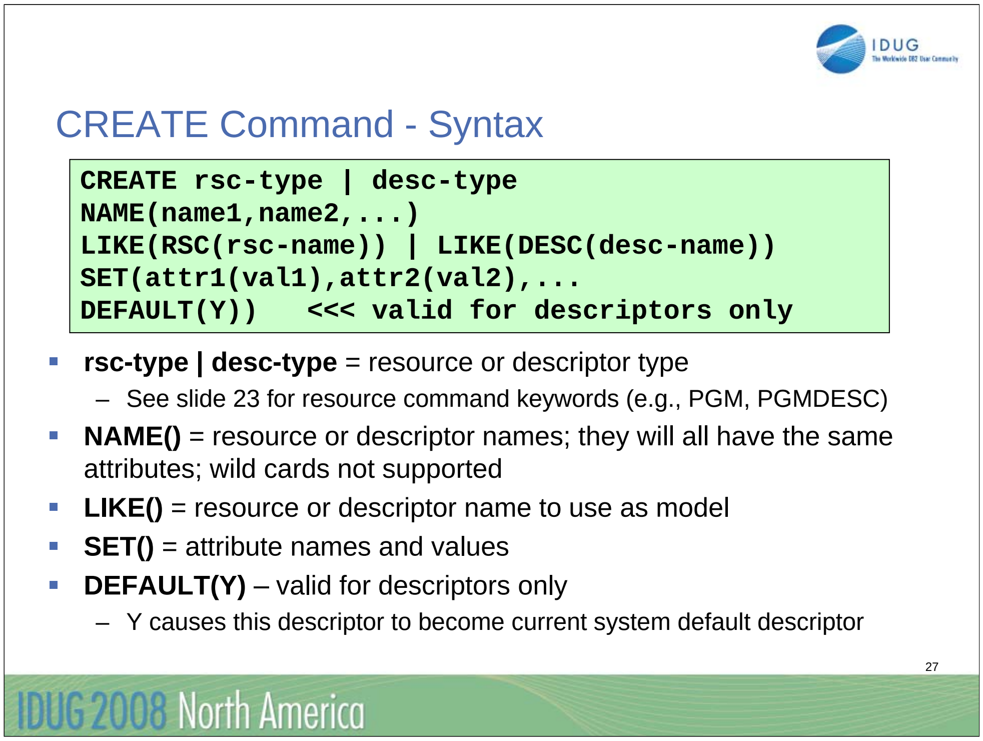
DRD Command Entry and Response

- In some cases, an “Error Text” will also be returned
- *IMS V10 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)

- Sample command input and response

```
DELETE DB NAME(ACCTMSTR,CUSTMSTR)
DBName     MbrName   CC   CCText                      ErrorText
ACCTMSTR   IMS1      11   REFERENCED BY PROGRAM       ACCTUPDT
CUSTMSTR   IMS1       0
```

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 is 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
  - See slide 23 for resource command keywords (e.g., PGM, PGMDESC)
- **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 system default descriptor

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

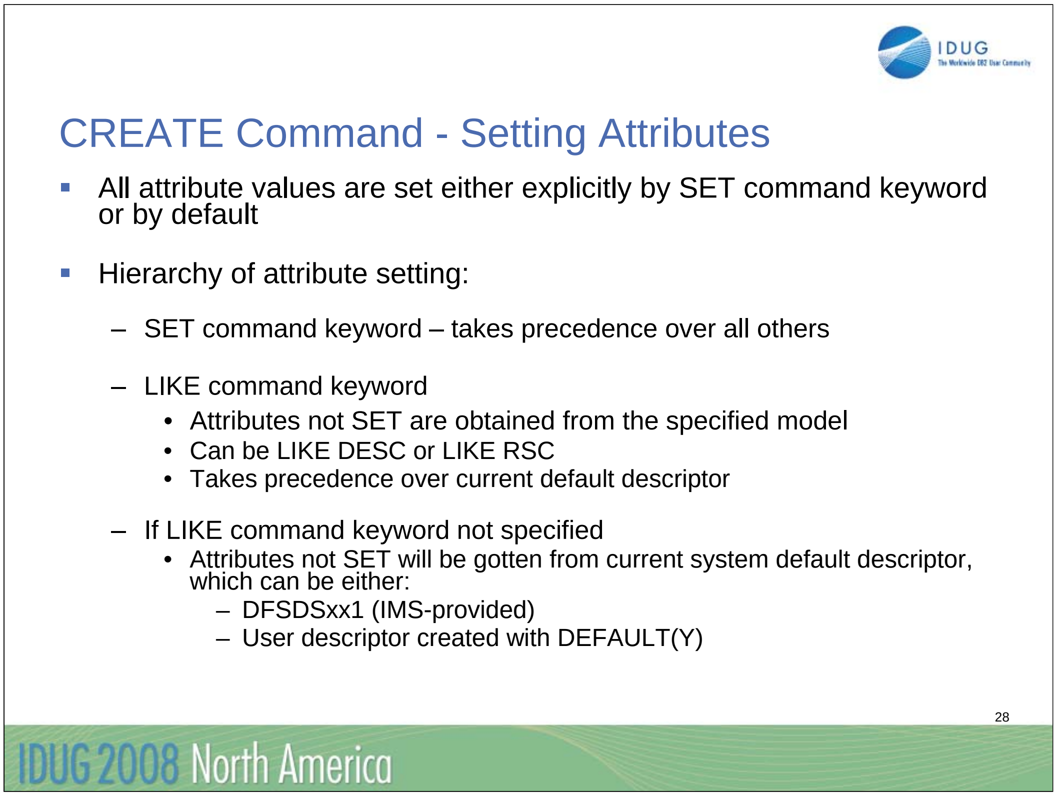
-rsc-type or desc-type – a previous slide showed the possibilities a DB, DBDESC, PGM, PGMDESC, TRAN, TRANDESC, RTC, RTCDESC.

-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 gotten from current system default descriptor, which can be either:
      - DFSDSxx1 (IMS-provided)
      - User descriptor created with DEFAULT(Y)

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.

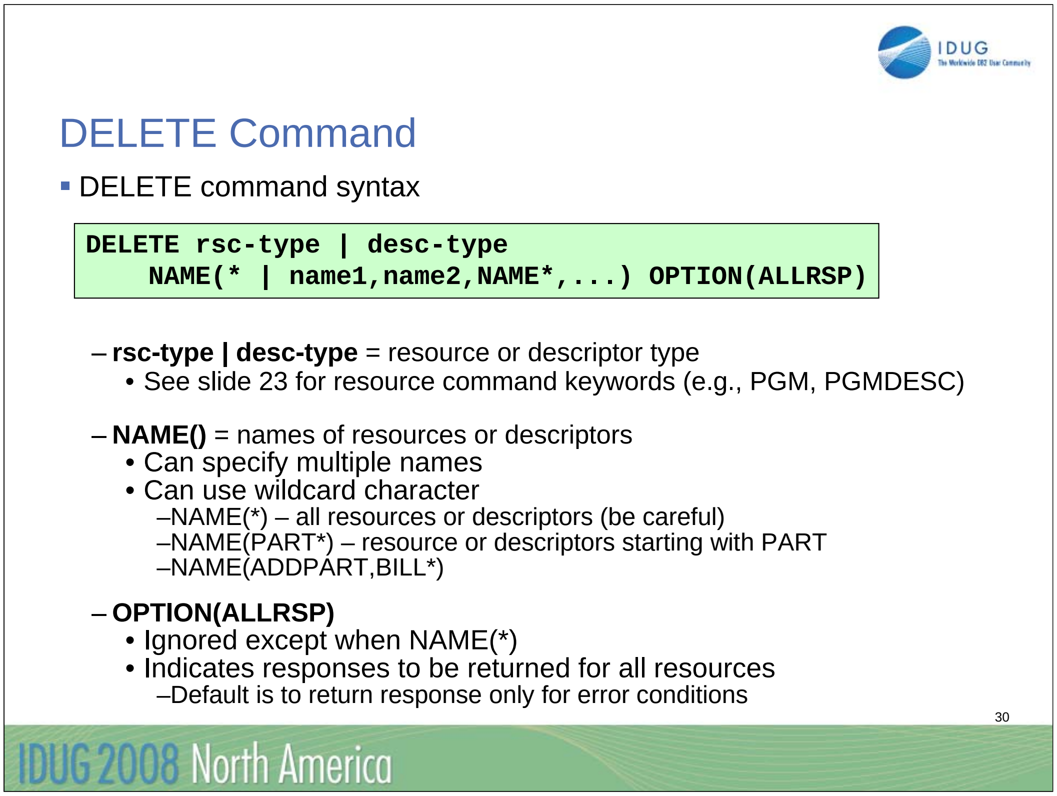
CREATE Command - Examples

- Creating a transaction with the IMS default descriptor for transactions
  - DFSDSTR1 is used since SET or LIKE not specified
  - All attributes defined such as:
    - PARLIM = 65535
    - PLCT = 65535
    - PLCTTIME = 6553500
    - MAXRGN = 0

```
CRE TRAN NAME(BALCHECK)
```

- 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))
```



DELETE Command

- DELETE command syntax

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

- **rsc-type | desc-type** = resource or descriptor type
  - See slide 23 for resource command keywords (e.g., PGM, PGMDESC)
- **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

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:

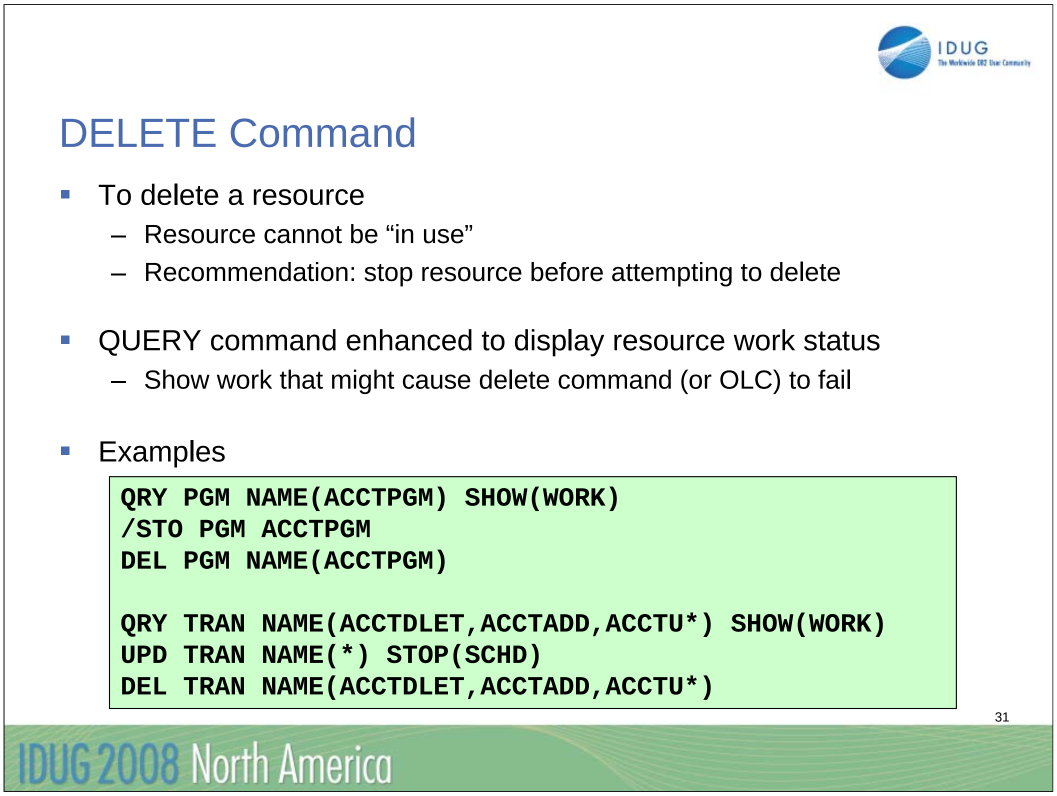
-“*” - which means multiple character substitution; it can be used only at the end of the name (e.g., PART*)

-“!” - which is single character substitution and can be anywhere within the name (e.g., PA!TMST)

-They may be combined in a single name (e.g., PA!T*)

-Except for “*” 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



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
- Examples

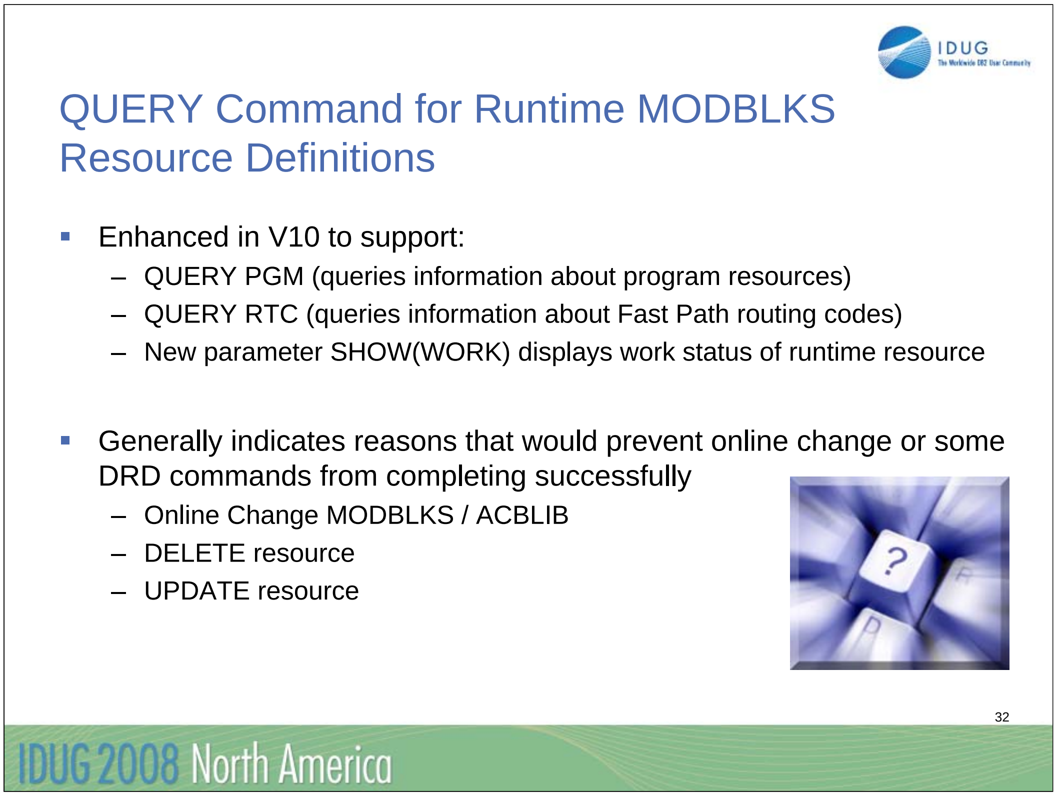
```
QRY PGM NAME(ACCTPGM) SHOW(WORK)
/STO PGM ACCTPGM
DEL PGM NAME(ACCTPGM)

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

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.

The **DELETE TRAN** command fails if any work is in progress for the transaction. The transaction is not stopped during **DELETE TRAN** command processing, so there is no automatic prevention of further queuing to the transaction. This is why it is important to stop a resource before attempting to delete it. The same holds true for the other MODBLKS resource types: the **DELETE PGM** command fails if any work is in progress for the program. The program is not stopped during **DELETE** command processing, and there it no automatically prevention of further scheduling of the program. The **DELETE RTC** command fails if any work is in progress for the routing code. The routing code is not is stopped during **DELETE RTC** command processing, and there is no automatically prevention further queuing to the routing code. You can stop a resource with a /STOP command, or by using the UPDATE command with a STOP(SCHD) parameter included.

You can QUERY the resource before attempting to delete it to determine whether there is



QUERY Command for Runtime MODBLKS Resource Definitions

- Enhanced in V10 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
- Generally indicates reasons that would prevent online change or some DRD commands from completing successfully
  - Online Change MODBLKS / ACBLIB
  - DELETE resource
  - UPDATE resource

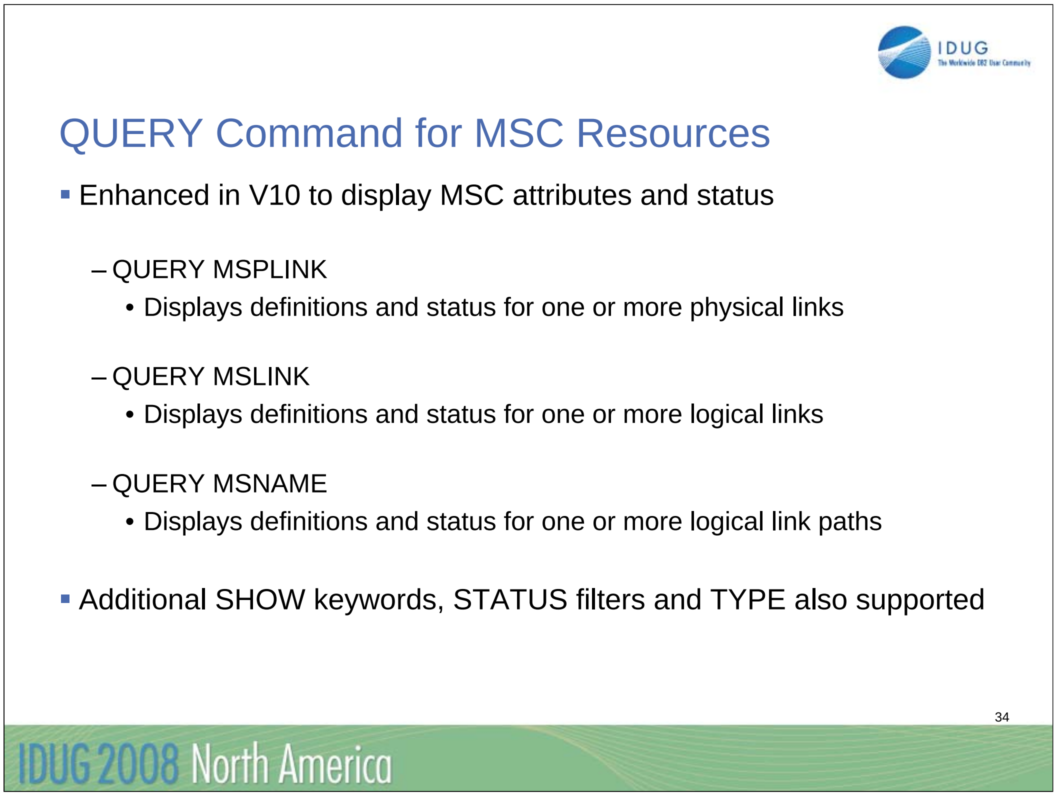
The QUERY command can be issued with the SHOW(WORK) parameter specified to check for any work in progress for the specified resources and other resources that are associated with it. Issuing this command before a DELETE, IMPORT or UPDATE command is useful since it would show you any work in progress might cause these DELETE, IMPORT 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.

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 Runtime MODBLKS Resource Definitions - Examples

```
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)
```



QUERY Command for MSC Resources

- Enhanced in V10 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
- Additional SHOW keywords, STATUS filters and TYPE also supported

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 V10 introduces the ability to query MSC definitions and a few examples are listed here. For example, you can query an MSPLINK which is the physical MSC link named IMSAB to determine its logical link and that logical link's path. You can query an MSLINK, which is the

MSLINK

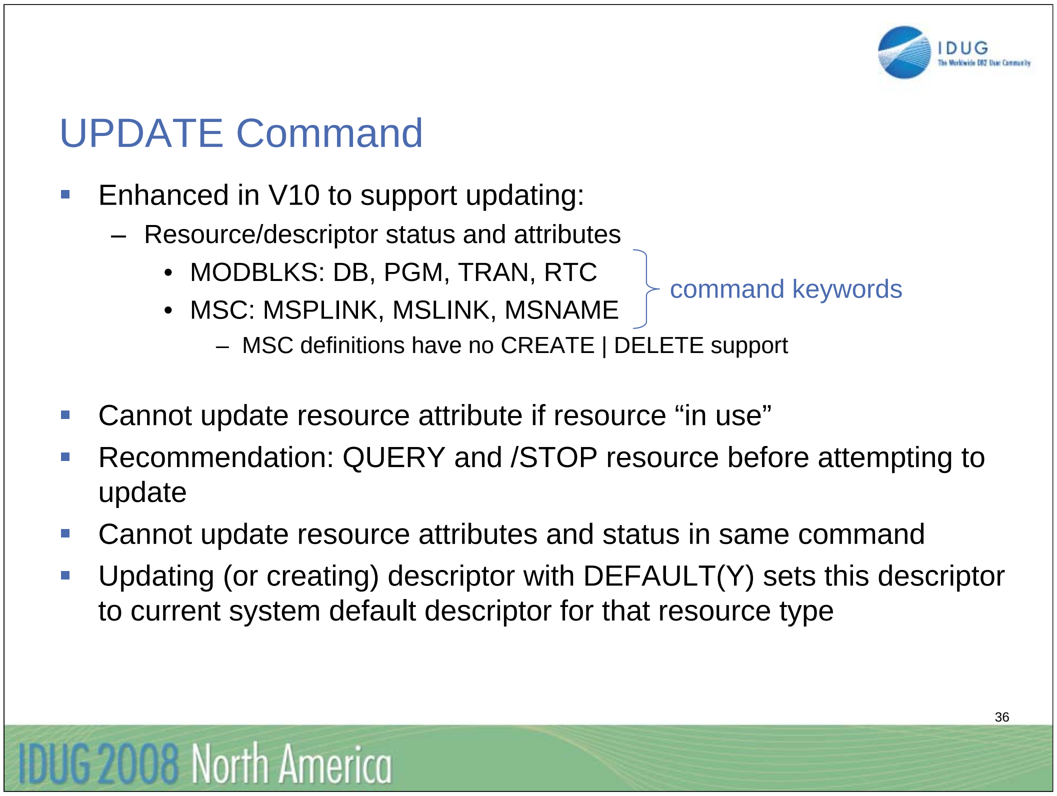
Displays the logical links that are associated with the specified physical link. When the physical link is VTAM, the maximum number of allowed logical sessions for the physical link is also displayed. When SHOW(MSLINK) is specified, STATUS() is invalid. SHOW(MSNAME) and SHOW(MSLINK) are mutually exclusive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

QUERY Command for MSC Resources - Examples

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

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

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



UPDATE Command

- Enhanced in V10 to support updating:
  - Resource/descriptor status and attributes
    - MODBLKS: DB, PGM, TRAN, RTC
    - MSC: MSPLINK, MSLINK, MSNAME
      - MSC definitions have no CREATE | DELETE support

command keywords

- 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

The UPDATE command has been enhanced significantly in V10, 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 V10, 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 condition 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 system default. This would only be needed if you first made a user-defined descriptor the default and then later wanted to restore DFSDSxx1 and the default.

The UPDATE command has been enhanced significantly in V10, 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 V10, 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 condition code (a number) and

UPDATE Command

- 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))
```

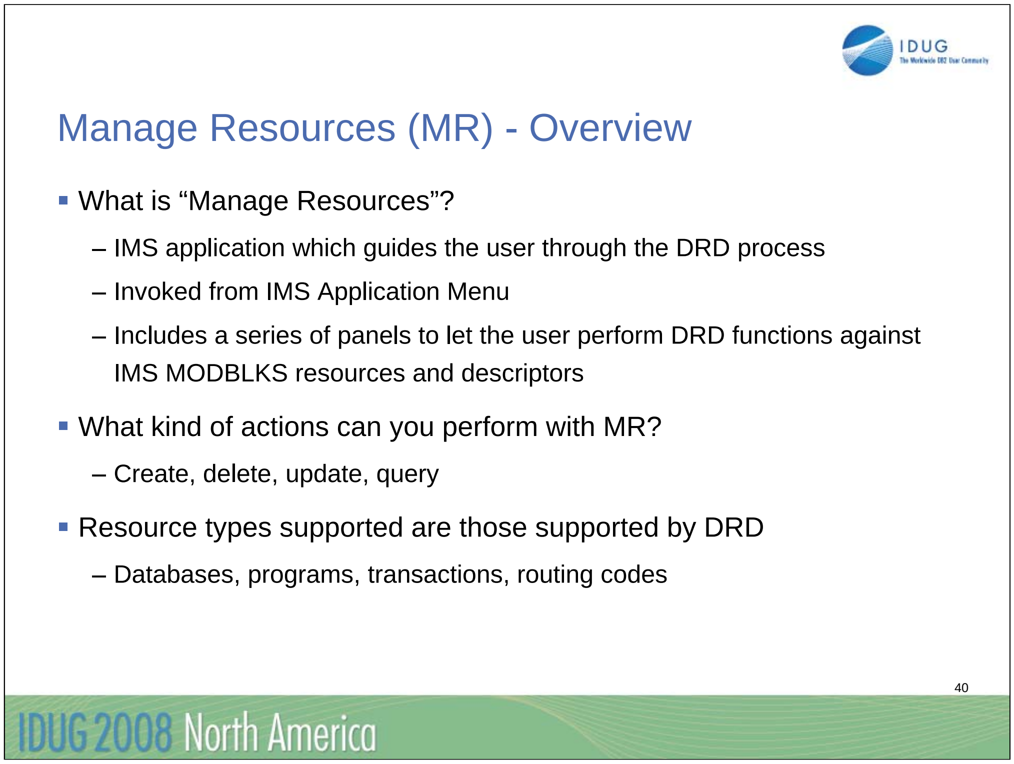
UPDATE Command for MSC - Examples

- 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

Manage Resources Application

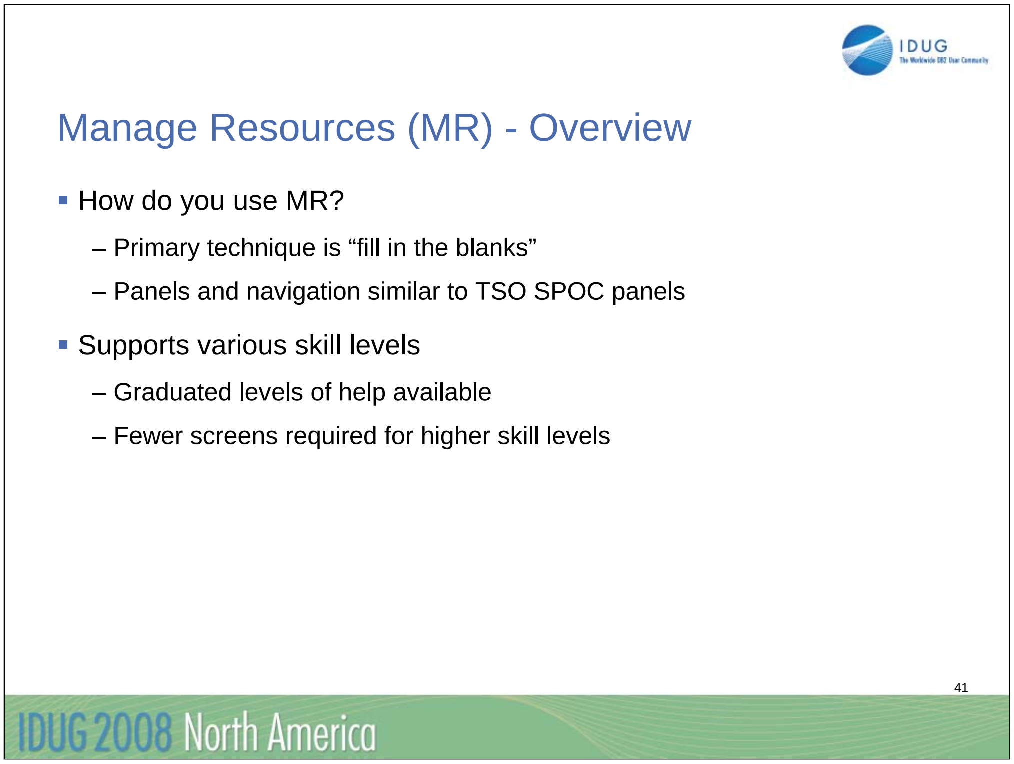


Manage Resources (MR) - Overview

- 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
- Resource types supported are those supported by DRD
  - – Databases, programs, transactions, routing codes

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.

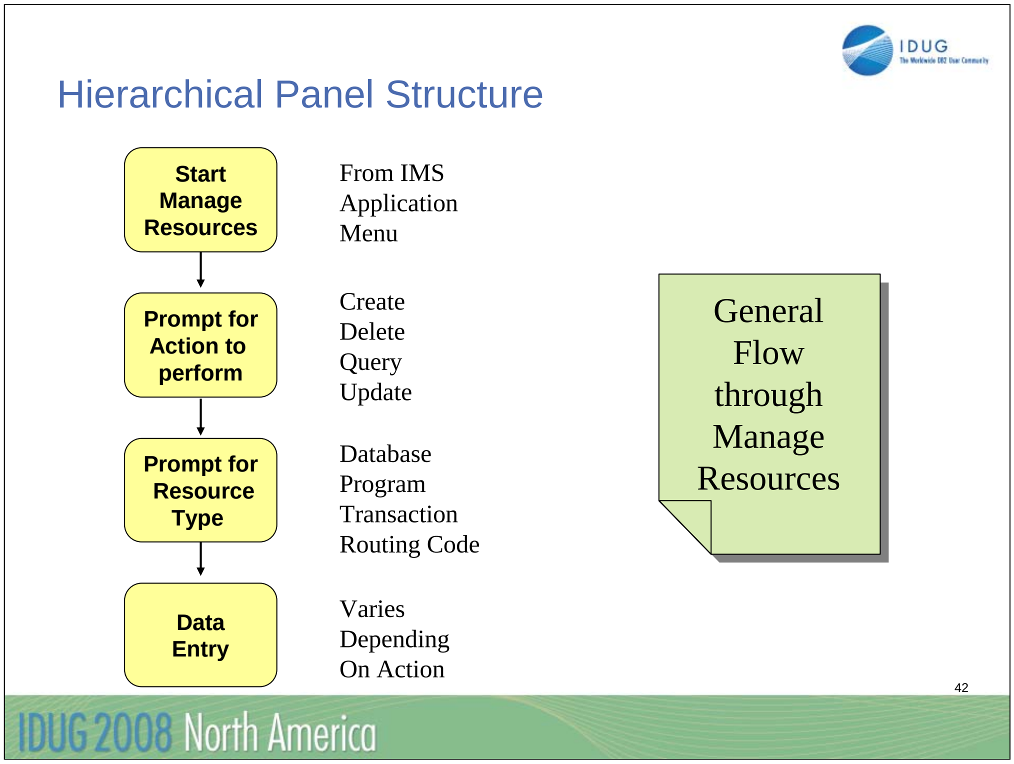


Manage Resources (MR) - Overview

- 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

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

Start Manage Resources

From IMS Application Menu

Prompt for Action to perform

Create
Delete
Query
Update

Prompt for Resource Type

Database
Program
Transaction
Routing Code

Data Entry

Varies Depending On Action

General Flow through Manage Resources

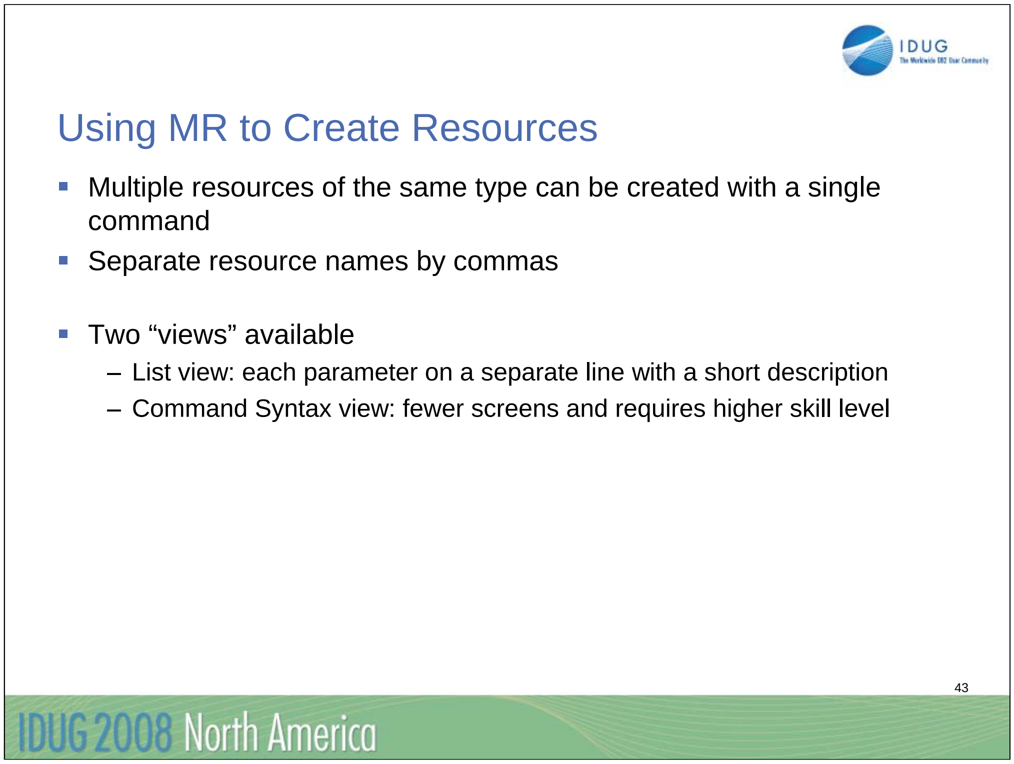
The basic flow through MR is:

-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)

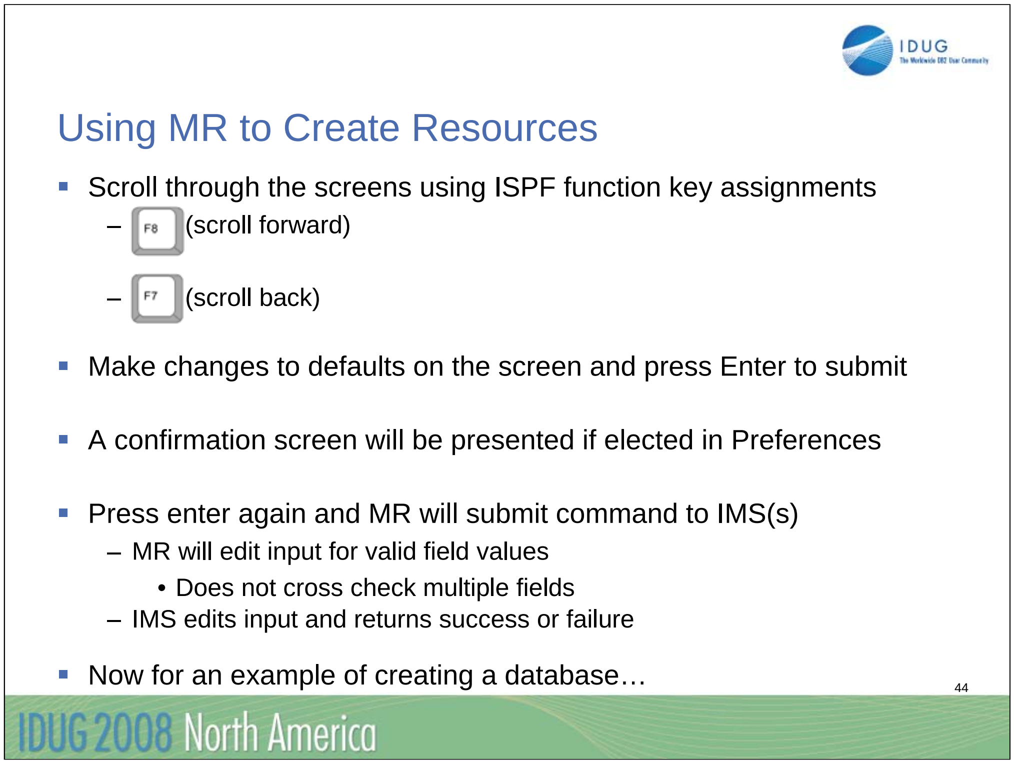
-Enter the data (e.g., enter the database name and attributes)



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

The next few screens are an example of creating a transaction descriptor, then using that descriptor to create two transactions, then creating a database. We will show both the List View and Command Syntax View.



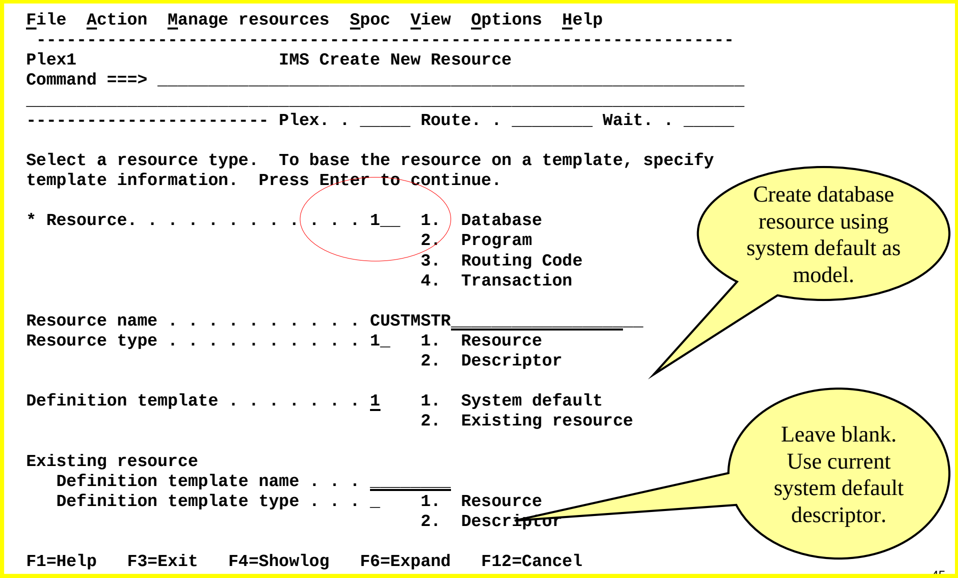
Using MR to Create Resources

- Scroll through the screens using ISPF function key assignments
  - F8 (scroll forward)
  - F7 (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…

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.

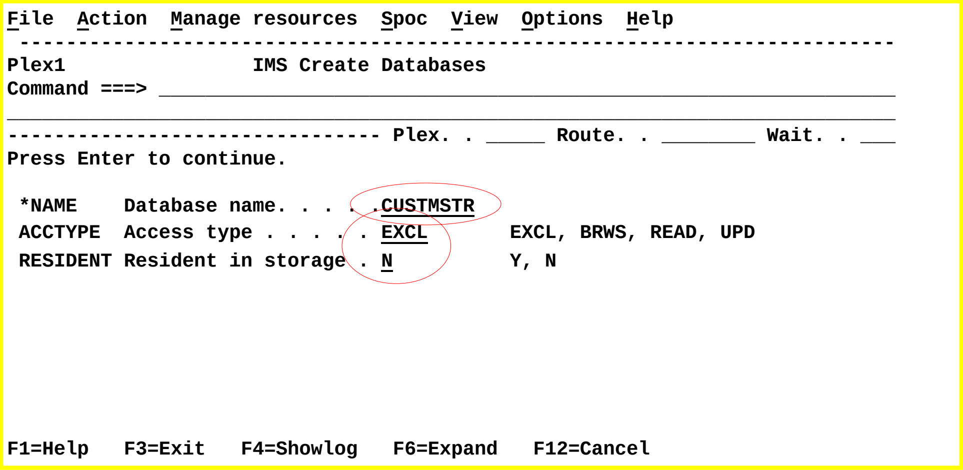
Now lets create a database. Here we say we want to create a

-database

-resource

-using the current system transaction descriptor for fields values we don’t supply

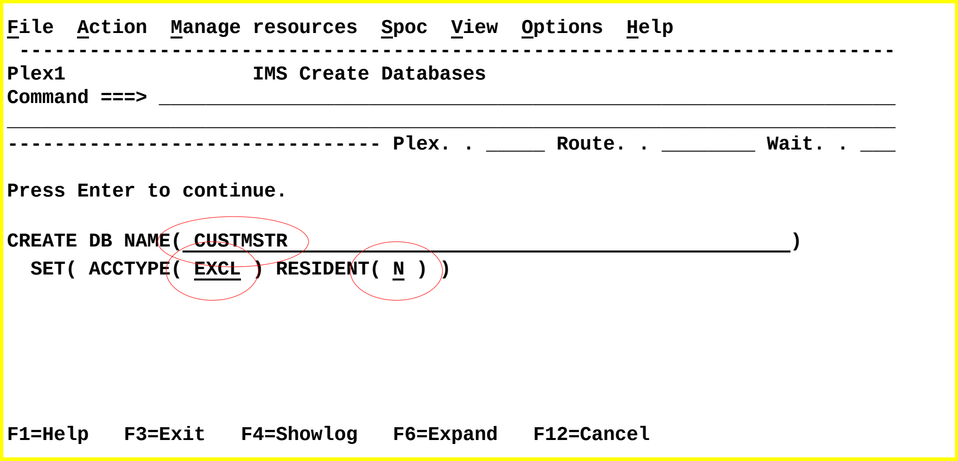
MR’s List View of Creating a Database



```
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
```

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

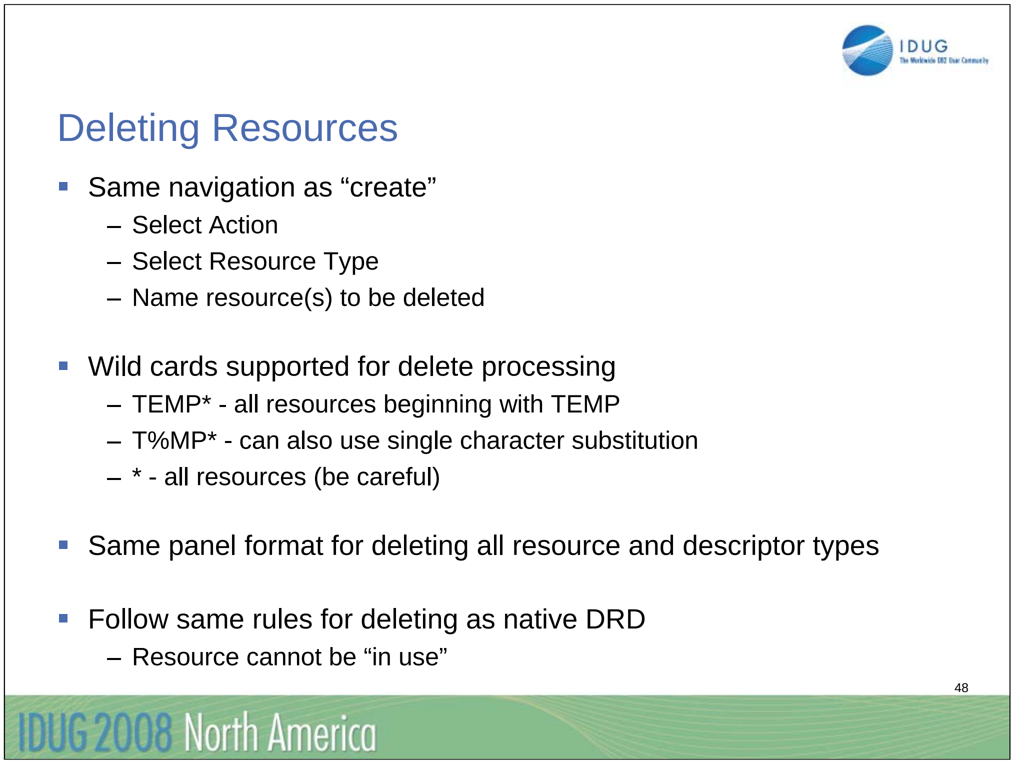


```
File  Action  Manage resources  Spoc  View  Options  Help
 ----------------------------------------------------------------------------
Plex1                    IMS Create Databases
Command ===> _____________________________________________________________
_____________________________________________________________________________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

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”

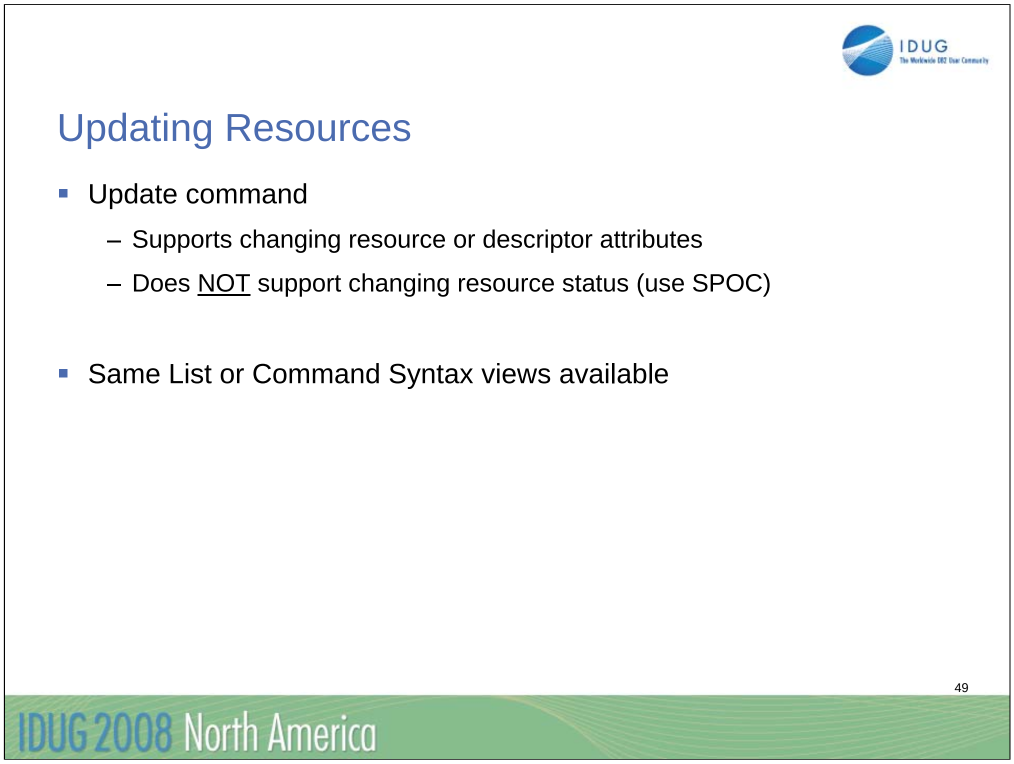
Deleting resources or descriptors follows the same general navigation path

-select an action (delete)

-select a type

-identify (name) the resource

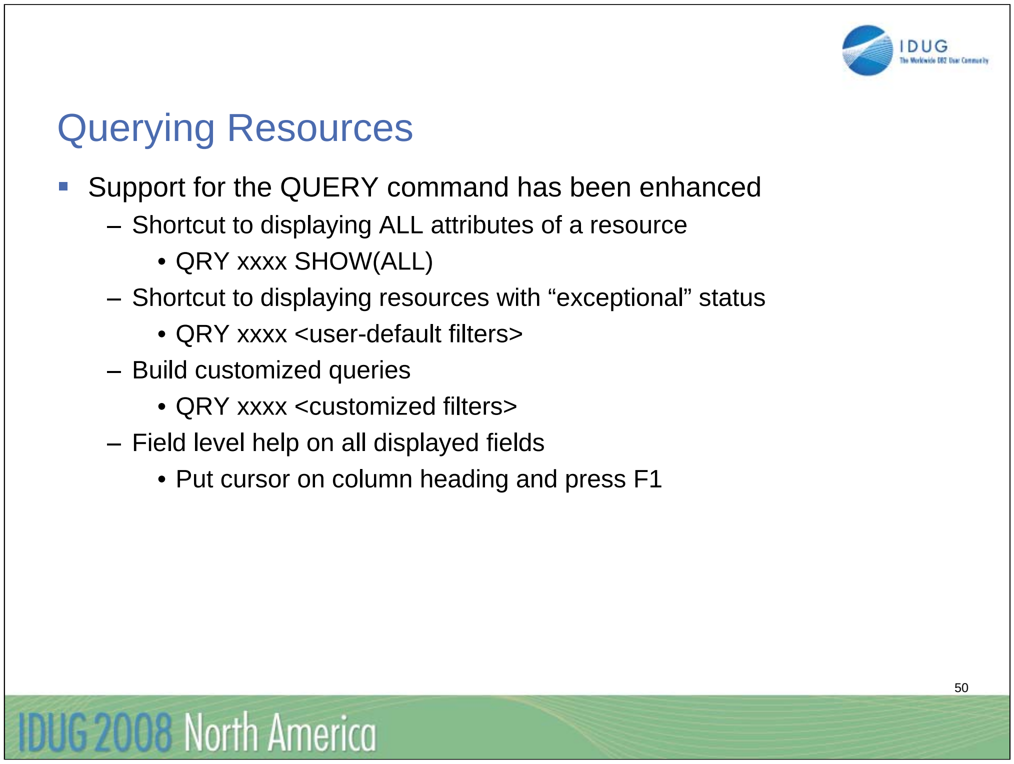
For delete, wild cards are supported, including “*” which is delete ALL – be careful.



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

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

MR also supports an enhanced QRY capability.

-a shortcut for displaying all attributes of a resource

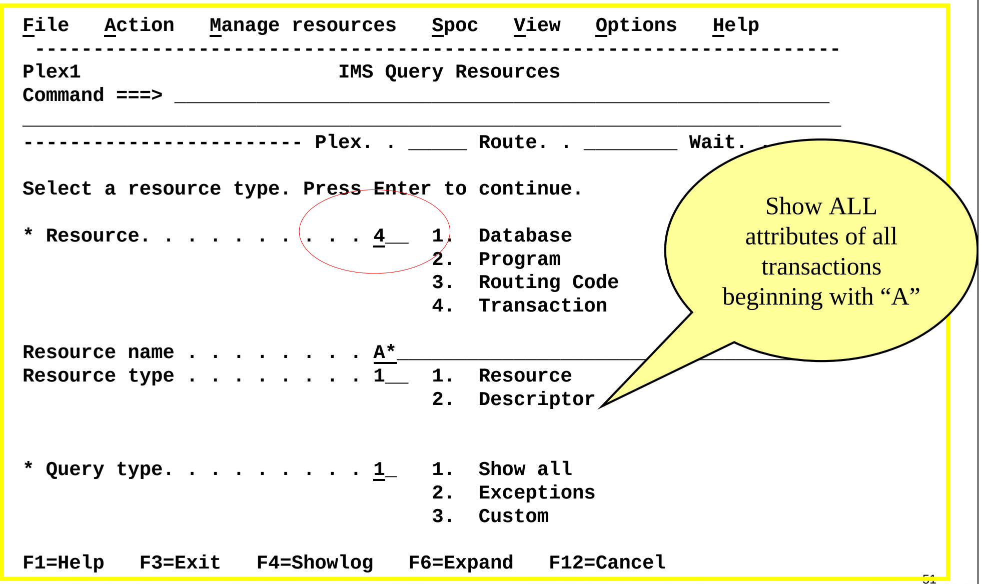
-a shortcut for displaying resources with “exceptional” status (You get to decide what is exceptional)

-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
```

Show ALL attributes of all transactions beginning with “A”

From this screen you can enter

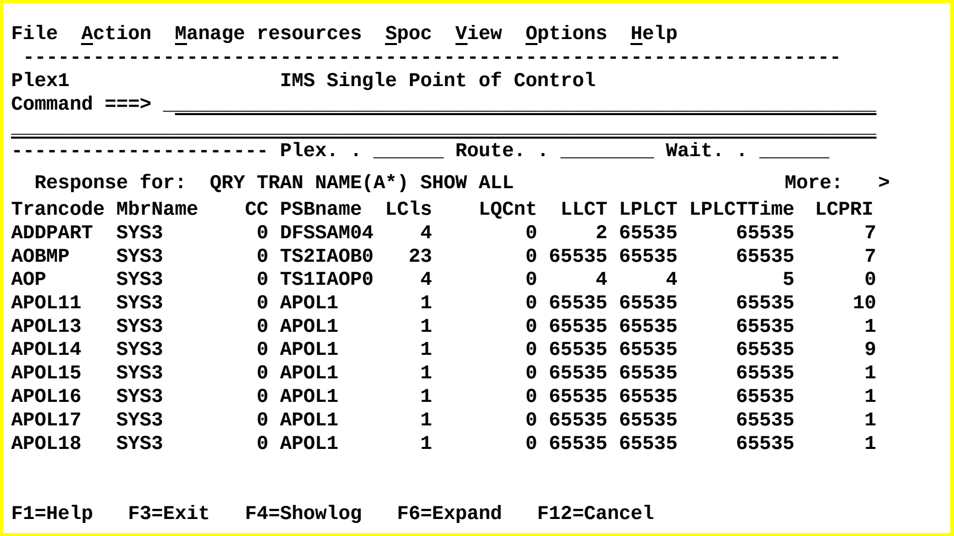
-resource type (transaction)

-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
```

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.

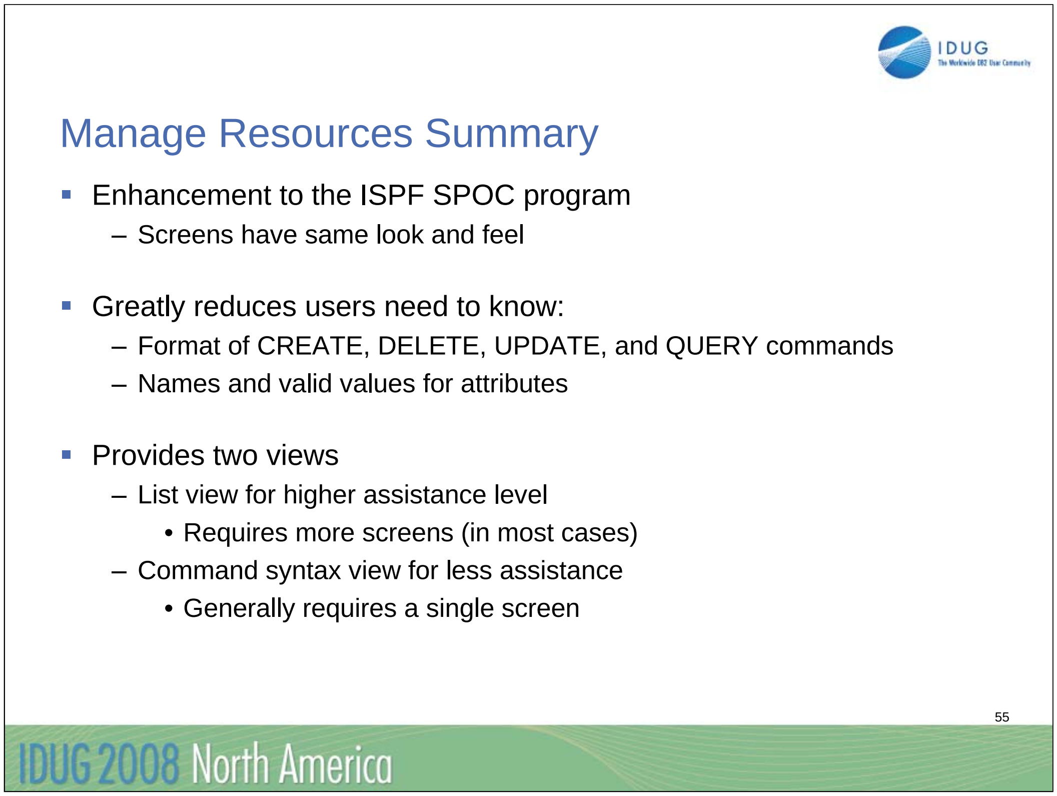
If you put a "/" in the action column, you can then pull down the Action Menu from the Action Bar at the top and select an action. That action will be executed for every resource selected.

Summary

Dynamic Resource Definition Summary

- IMS V10 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
  - Requires
    - Enabling parameters in DFSDFxxx PROCLIB member
    - CSL environment with SCI and OM
      - 1 SCI per z/OS image and 1 OM per IMSplex
    - Cold start

- MSC DRD allows dynamic updating of MSC definitions
  - Requires
    - CSL environment with SCI and OM



Manage Resources Summary

- Enhancement to the ISPF SPOC program
  - Screens have same look and feel
- Greatly reduces users need to know:
  - Format of CREATE, DELETE, UPDATE, and QUERY commands
  - Names and valid values for attributes
- Provides two views
  - List view for higher assistance level
    - Requires more screens (in most cases)
  - Command syntax view for less assistance
    - Generally requires a single screen

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.

Questions

Session K06

Session IMS V10 Dynamic Resource Definition

Angelique Greenhaw

IBM Corporation

greenhaw@us.ibm.com

Thank you!