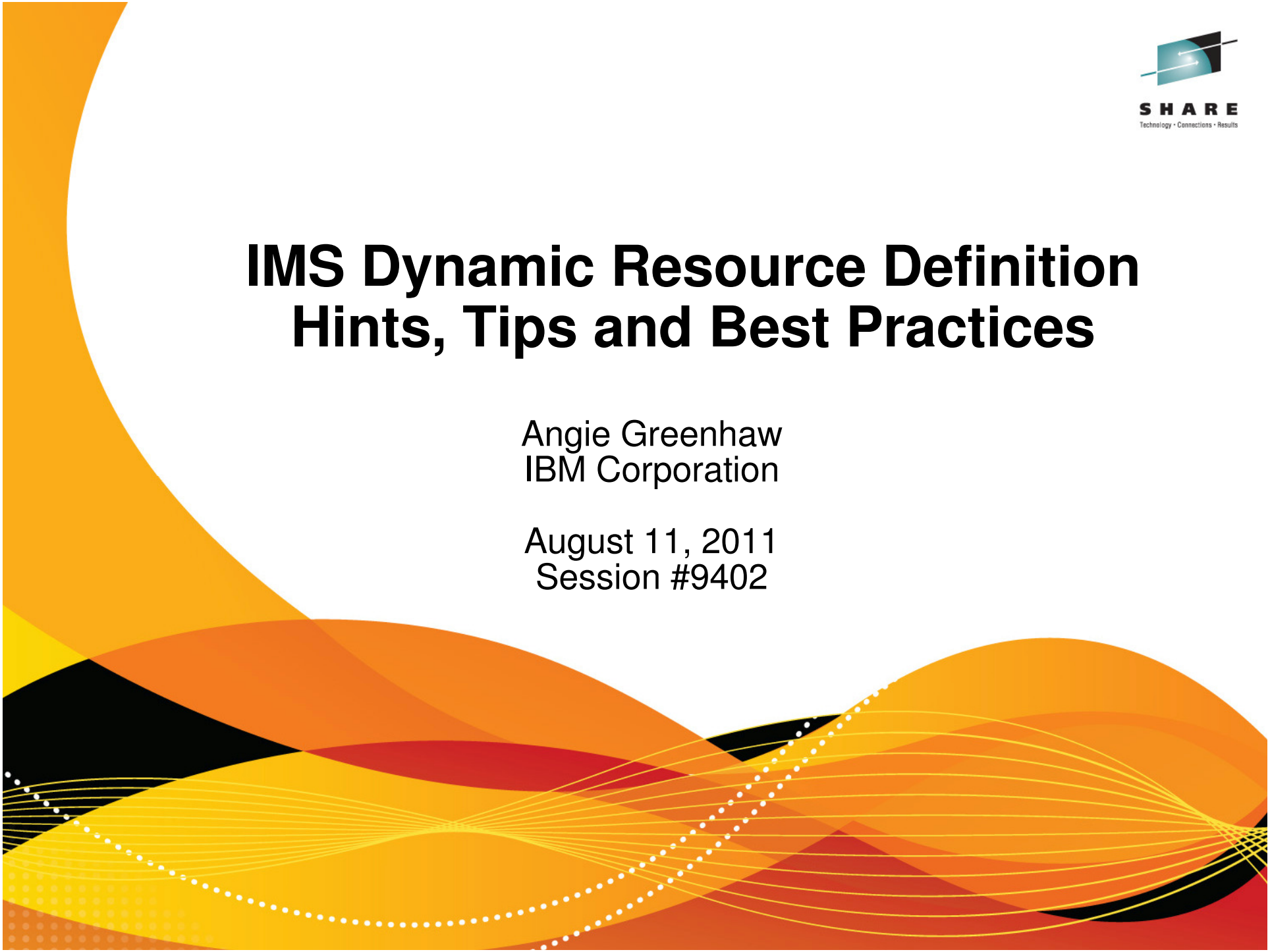


IMS Dynamic Resource Definition Hints, Tips and Best Practices

Angie Greenhaw
IBM Corporation

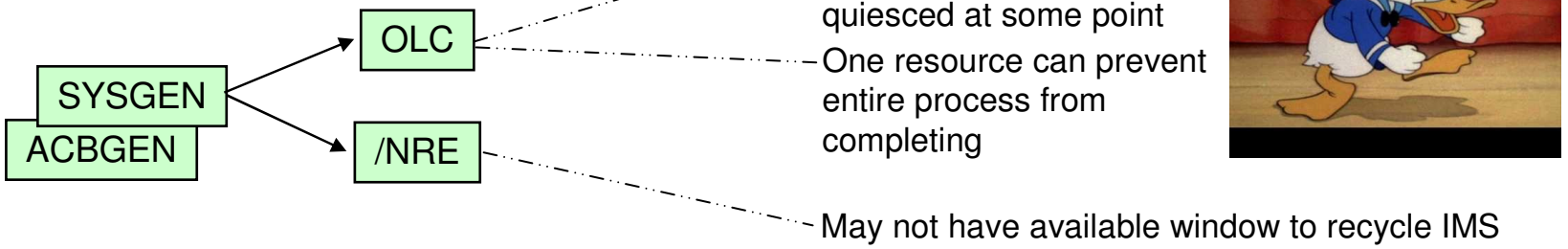
August 11, 2011
Session #9402

Agenda

- DRD overview
- DRD migration best practices
  - Data set management
  - Autoimport/Autoexport vs. IMPORT/EXPORT commands
  - Recommended parameter settings
- DRD utilities best practices
  - Leveraging utilities to help fold DRD into existing processes
- DRD usage best practices
  - Commands
  - Procedures
- DRD security considerations
- Summary/questions

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



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

DRD Overview

- OBJECTIVE: *Improve the availability* of the IMS online environment
- Allow user to *dynamically* define and enable *MODBLKS resource definitions*
  - *Databases*
  - *Programs*
  - *Transactions*
  - *Routing Codes*

DRD Overview

- Benefits
  - No requirement for MODBLKS SYSGEN
  - No requirement for IMS restart/MODBLKS online change
    - Limitations associated with these methods eliminated
  - Increased resource availability

DRD Overview

- Set of resource definition data sets (RDDS)
  - Contains statically (SYSGEN) and dynamically created definitions
    - Resource definitions
    - Model descriptors
- Type-2 commands: CREATE, DELETE, UPDATE, EXPORT, IMPORT
- Autoexport and autoimport functions provided to save/restore resource definitions across a cold start

DRD Overview

- 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

DRD Migration Best Practices

Managing Data Sets During Migration

- Keep MODBLKS data set and system RDDS synchronized during migration process
  - Why?
    - Enables fallback to OLC with MODBLKS data set in case DRD needs to be disabled
  - How?
    - Keep an up-to-date MODBLKS data set that matches most recently updated system RDDS
    - Use Extract RDDS Contents utility to generate STAGE1 macro definitions that reflect system RDDS contents
    - Generate MODBLKS data set with these STAGE1 definitions that are synchronized with the system RDDS

Managing Data Sets During Migration

- Keep MODBLKS data set and system RDDS synchronized during migration process
  - When?
    - Every time resource changes are made with DRD, keep MODBLKS data set up-to-date until migration process has been completed

Managing Data Sets After Migration

- Data set cleanup
  - System RDDS (most current) used for automatic import when IMS coldstarts unless MODBLKS is specified as the data set to import from
  - Delete MODBLKS data sets once DRD migration complete with successful testing

Recommended AUTOIMPORT Setting

- Specify AUTOIMPORT=AUTO in DFSDFxxx
  - IMS automatically determines which data set to import resource and descriptor definitions from (RDDS or MODBLKS data set)
  - Most current system RDDS selected if:
    - Two or more RDDSs are specified in DFSDFxxx
    - All defined RDDSs are allocated/readable
    - One contains valid IMS resource/descriptor definitions
  - MODBLKS data set selected if:
    - No system RDDSs are defined in DFSDFxxx or if they are empty
    - MODBLKS data set present, containing valid IMS resource/descriptor definitions

Recommended AUTOIMPORT Setting

- Specify AUTOIMPORT=AUTO in DFSDFxxx
  - Dynamically deleted resources with DELETE command will reappear in IMS system if imported from MODBLKS data set at next coldstart due to AUTOIMPORT=MODBLKS

Recommended AUTOEXPORT Setting

- Specify AUTOEXPORT=AUTO in DFSDFxxx
  - IMS will automatically export all resource and descriptor definitions to the oldest system RDDS at every system checkpoint
    - Definitional changes must have been made since the previous system checkpoint for this to occur
  - Ensures that definition updates have been captured and will be available for automatic import during next IMS coldstart

Populating System RDDS First Time

- AUTOIMPORT=AUTO and AUTOEXPORT=AUTO especially useful during initial migration
  - When coldstarting IMS for the first time after enabling DRD, its system RDDS will be empty, so IMS will autoimport definitions from MODBLKS dataset
  - Autoexport will occur after coldstart complete, populating the system RDDS with the definitions just read in from MODBLKS data set
  - At next coldstart, IMS will autoimport from the system RDDS since it now contains definitions

EXPORT/IMPORT commands

- EXPORT command allows for the exporting of MODBLKS resources and descriptor definitions to an RDDS
- IMPORT command allows for the importing of MODBLKS resources and descriptor definitions from an RDDS
- How can these commands help with porting resources from one IMS to another?

Application Migration with DRD

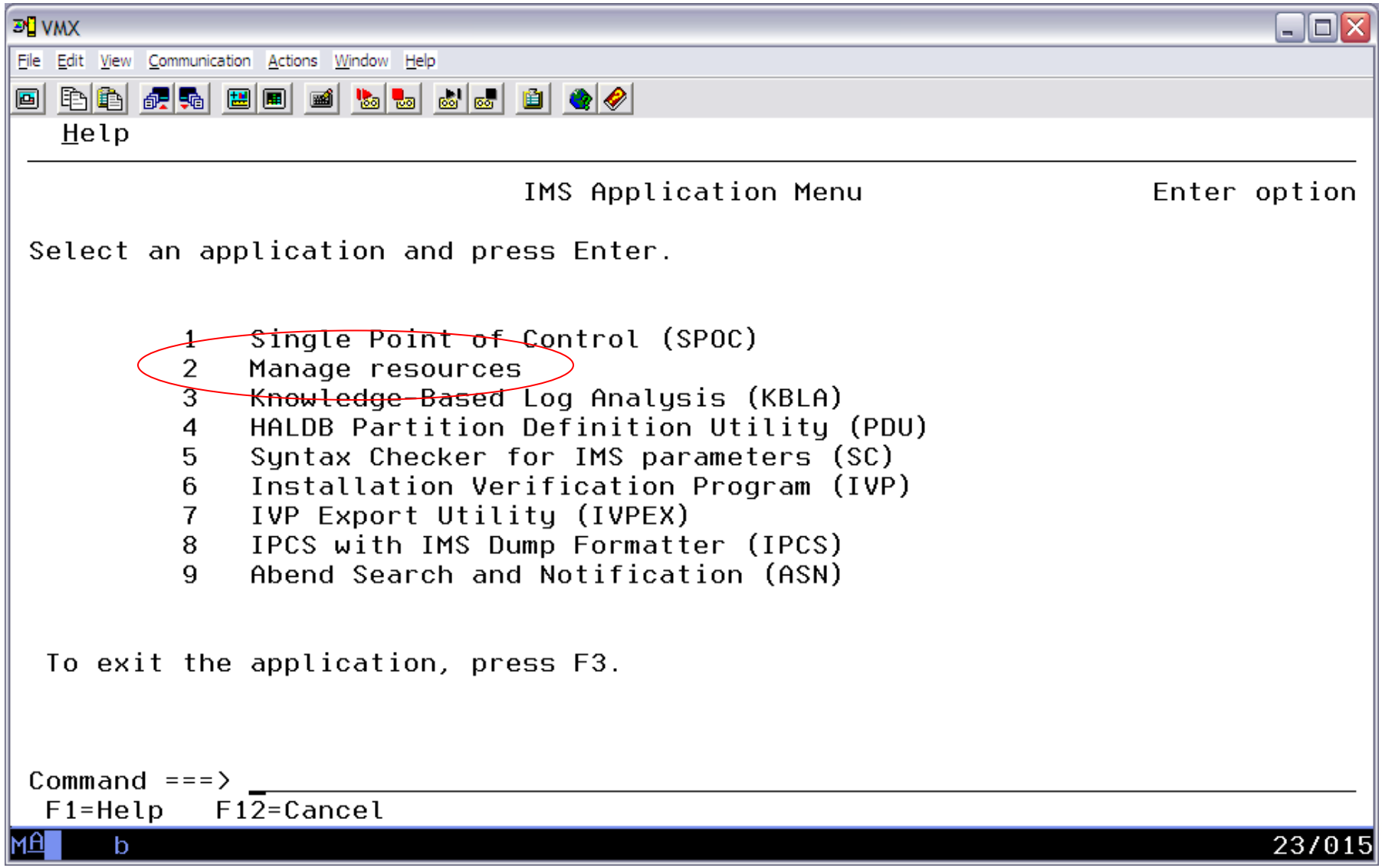
- Use EXPORT/IMPORT to migrate an application defined on one IMS system to another IMS system
- Issue EXPORT command on IMSA to export an application's database, program, routing code and transaction definitions to a non-system RDDS
- Issue IMPORT command on IMSB to import the definitions from the non-system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

Cloning IMS Systems with DRD

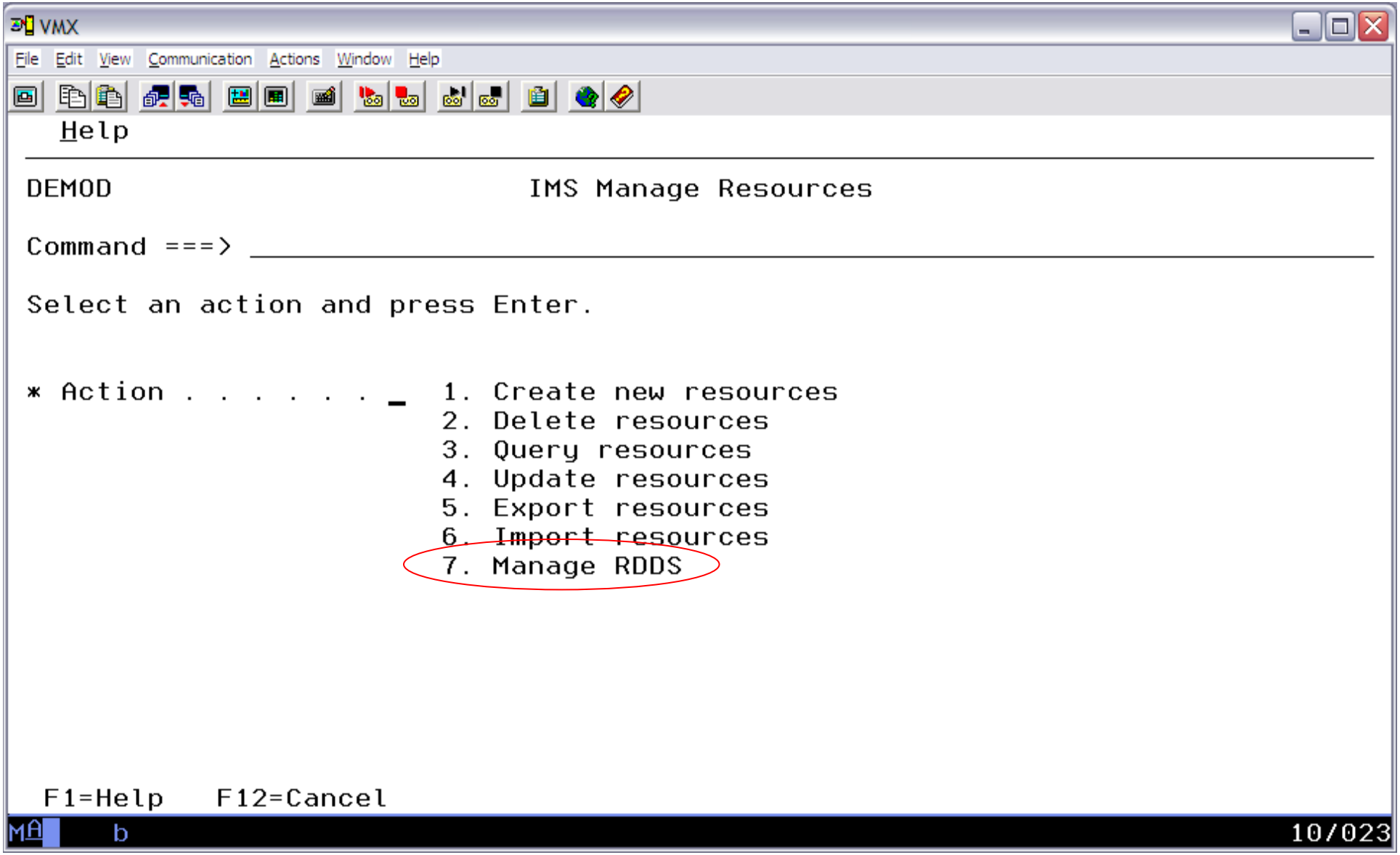
- Use EXPORT/IMPORT in these example steps
  - EXPORT all definitions from IMSA to non-system RDDS
  - Coldstart new IMSB with no resources defined
  - IMPORT definitions to IMSB from non-system RDDS
- Use Extract RDDS Contents utility to generate CREATE commands that match an IMS system's definitions
  - Coldstart new IMS system with no resources defined
  - Submit CREATE commands to this IMS using Batch SPOC utility
- Use Copy RDDS utility to copy contents to new IMS's system RDDS
  - Coldstart of new IMS system will read this populated system RDDS

DRD Utilities Best Practices

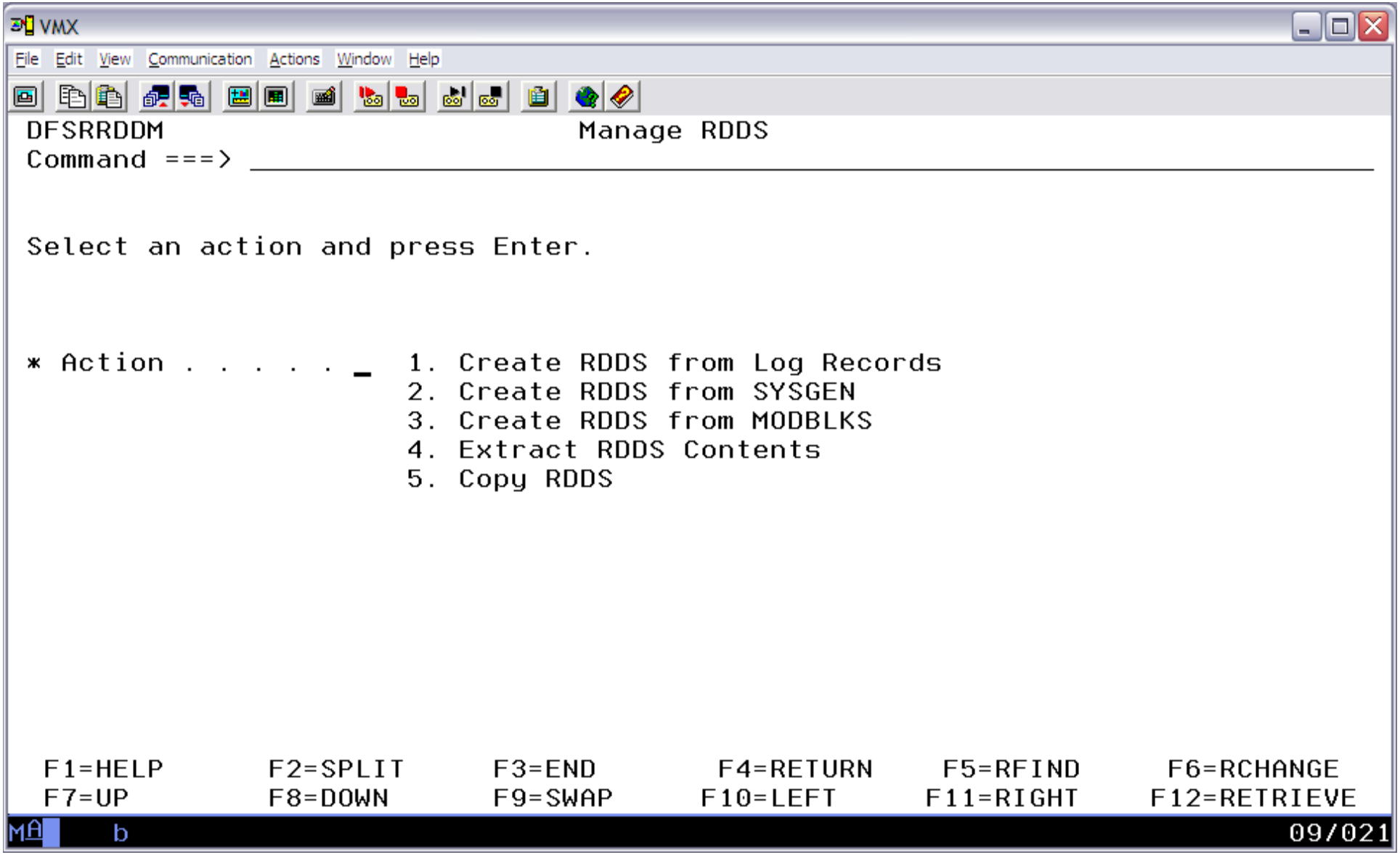
IMS Application Menu



Manage Resources Menu



Manage RDDS Menu



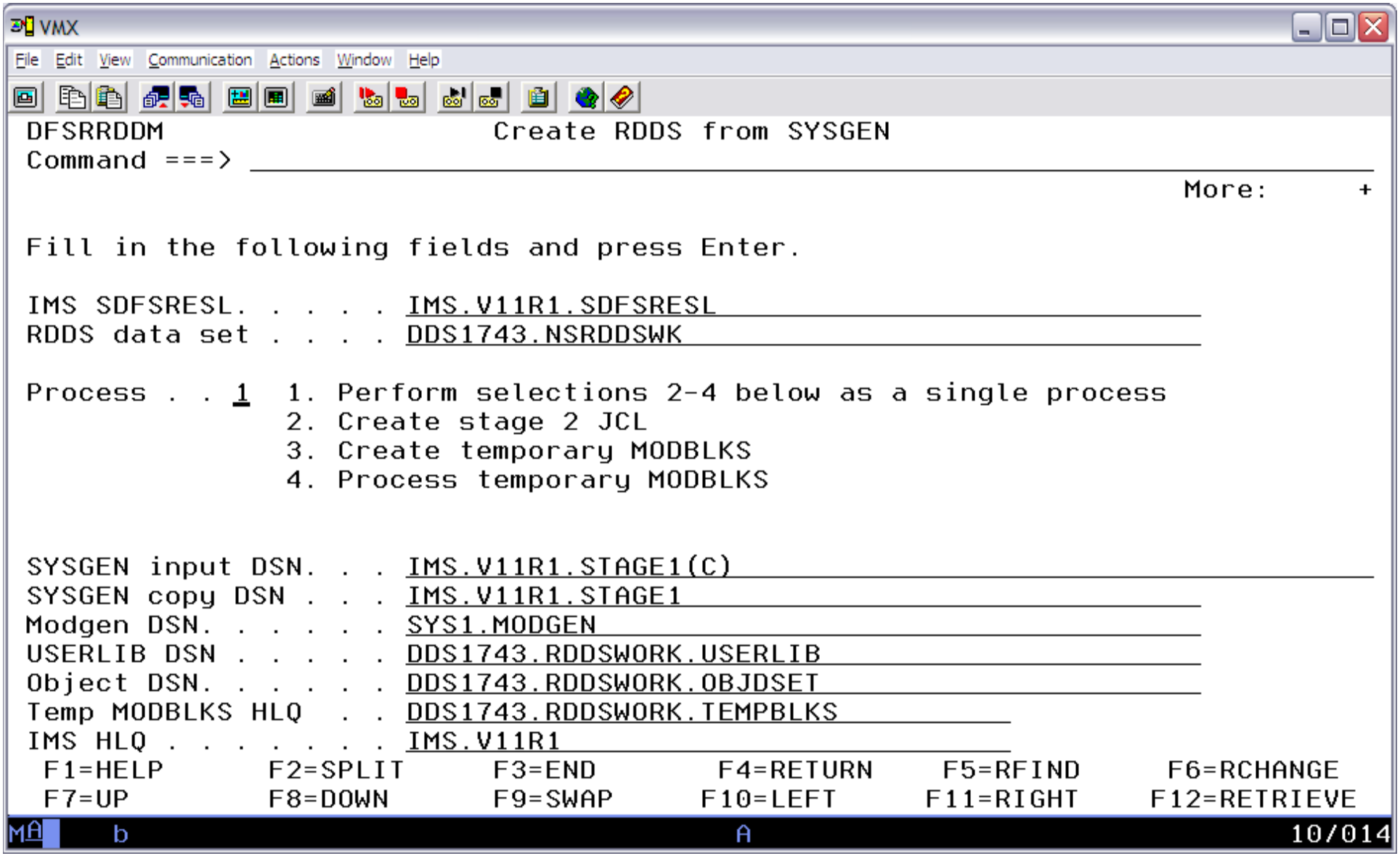
Utilities Help Fold DRD into Change Management Process

- Existing processes commonly involve
  - Extracting IMS resource data from a source
  - Generating SYSGEN macro statements
  - Running SYSGEN
  - Storing SYSGEN results in MODBLKS data set
  - Online change
- DRD utilities work with SYSGEN statements to aid in initial DRD migration, here's how…

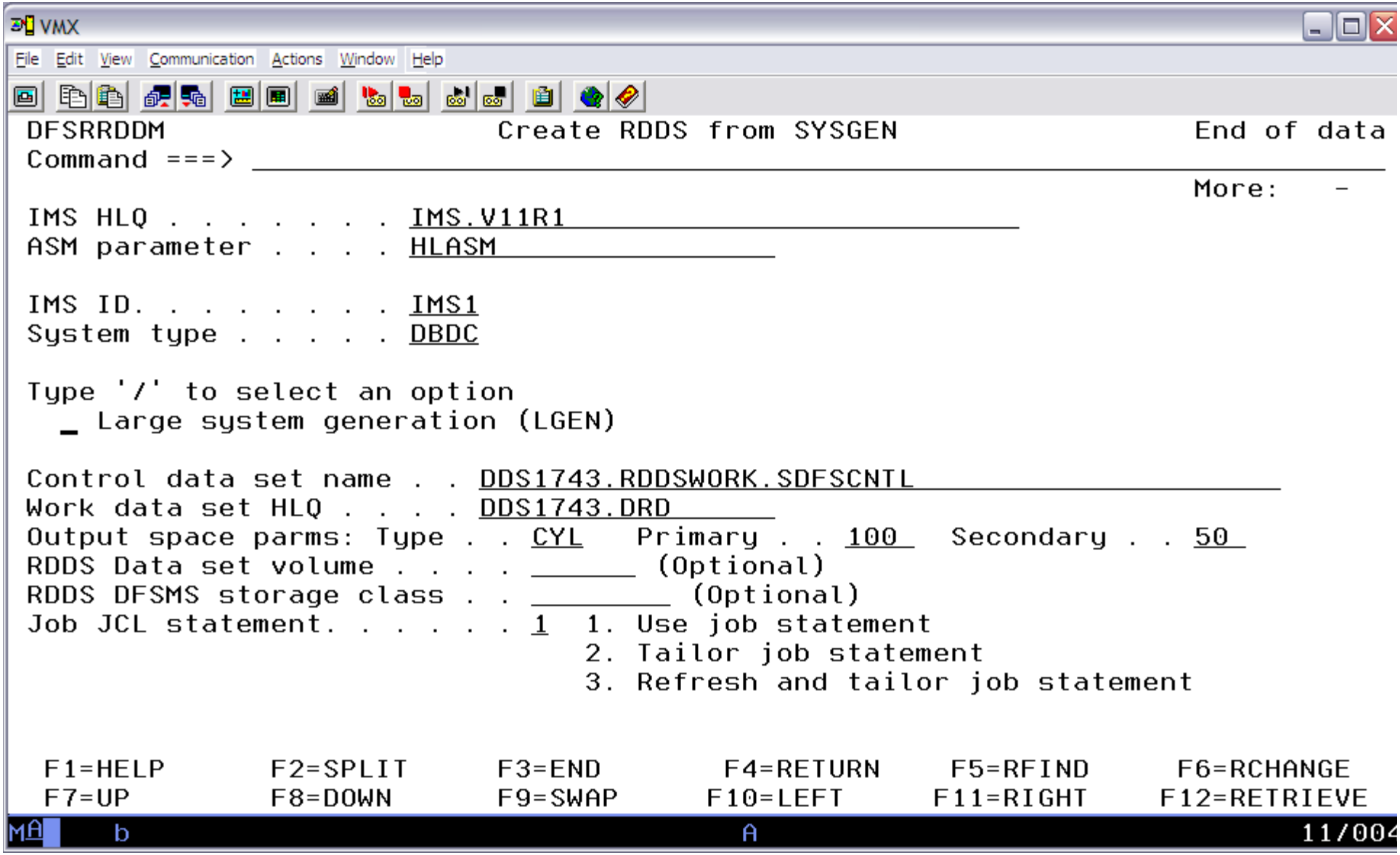
Using DRD Utilities with SYSGEN macros

- SYSGEN → RDDS
  - Run Create RDDS from SYSGEN utility to generate a system RDDS using SYSGEN macro statements as input
- SYSGEN → RDDS → CREATE commands
  - Complete previous step to generate RDDS
  - Run Extract RDDS Contents utility to generate CREATE commands using the newly generated RDDS as input
  - Can use Batch SPOC utility to submit CREATE commands to IMS

Create RDDS from SYSGEN Panel



Create RDDS from SYSGEN Panel (PF8)



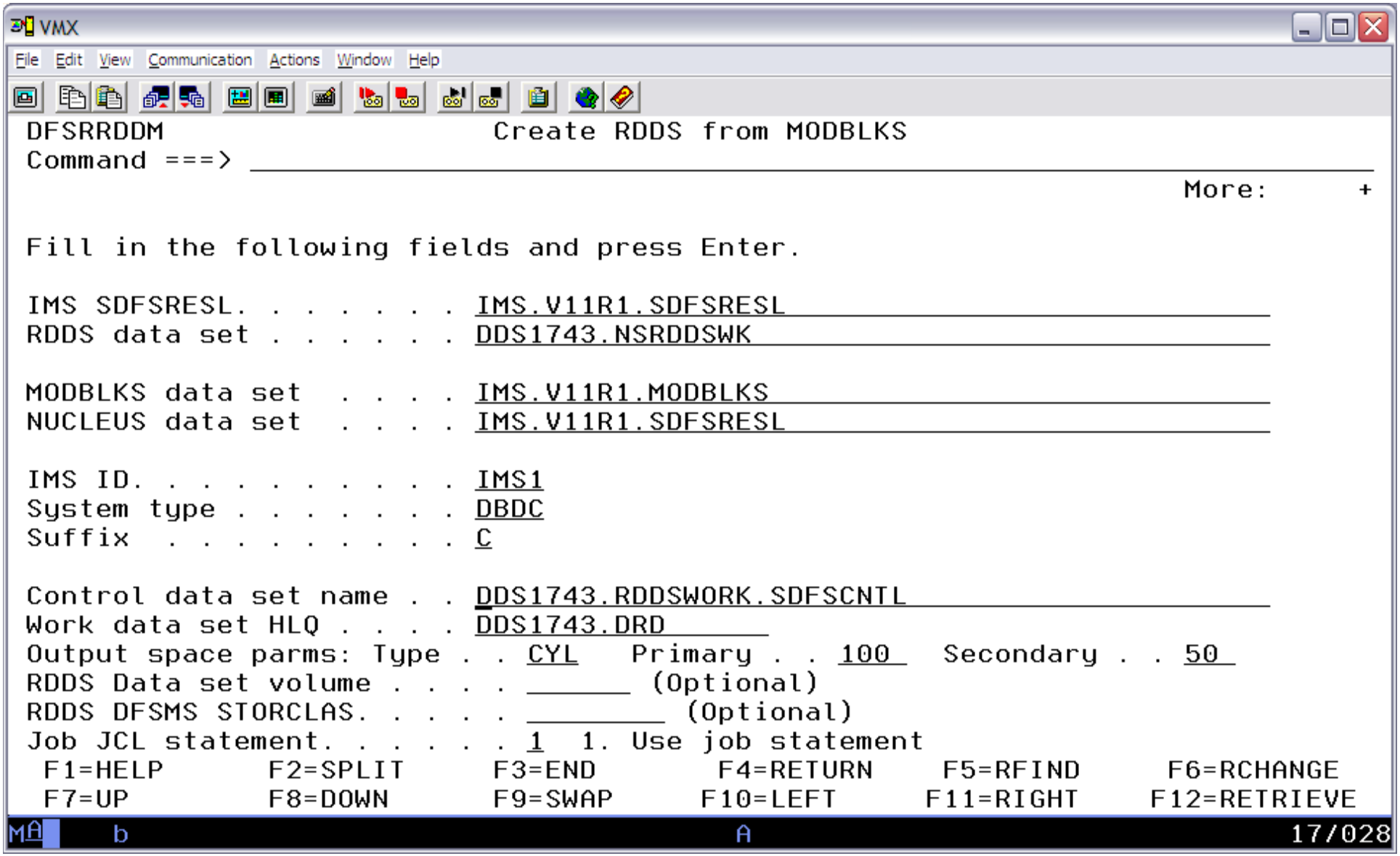
Utilities Leverage Existing Data to Help Migrate to DRD

- MODBLKS data set → RDDS or CREATE commands
- Log records → RDDS or CREATE commands
  - Use in a test/sandbox environment
  - Provides way of testing with actual data that matches development/production IMS system

Using DRD Utilities with MODBLKS Data Set

- MODBLKS → RDDS
  - Run Create RDDS from MODBLKS utility to generate an RDDS using MODBLKS data set input
- MODBLKS → RDDS → CREATE commands
  - Complete above steps to generate RDDS
  - Run Extract RDDS Contents utility to generate CREATE commands using the newly generated RDDS

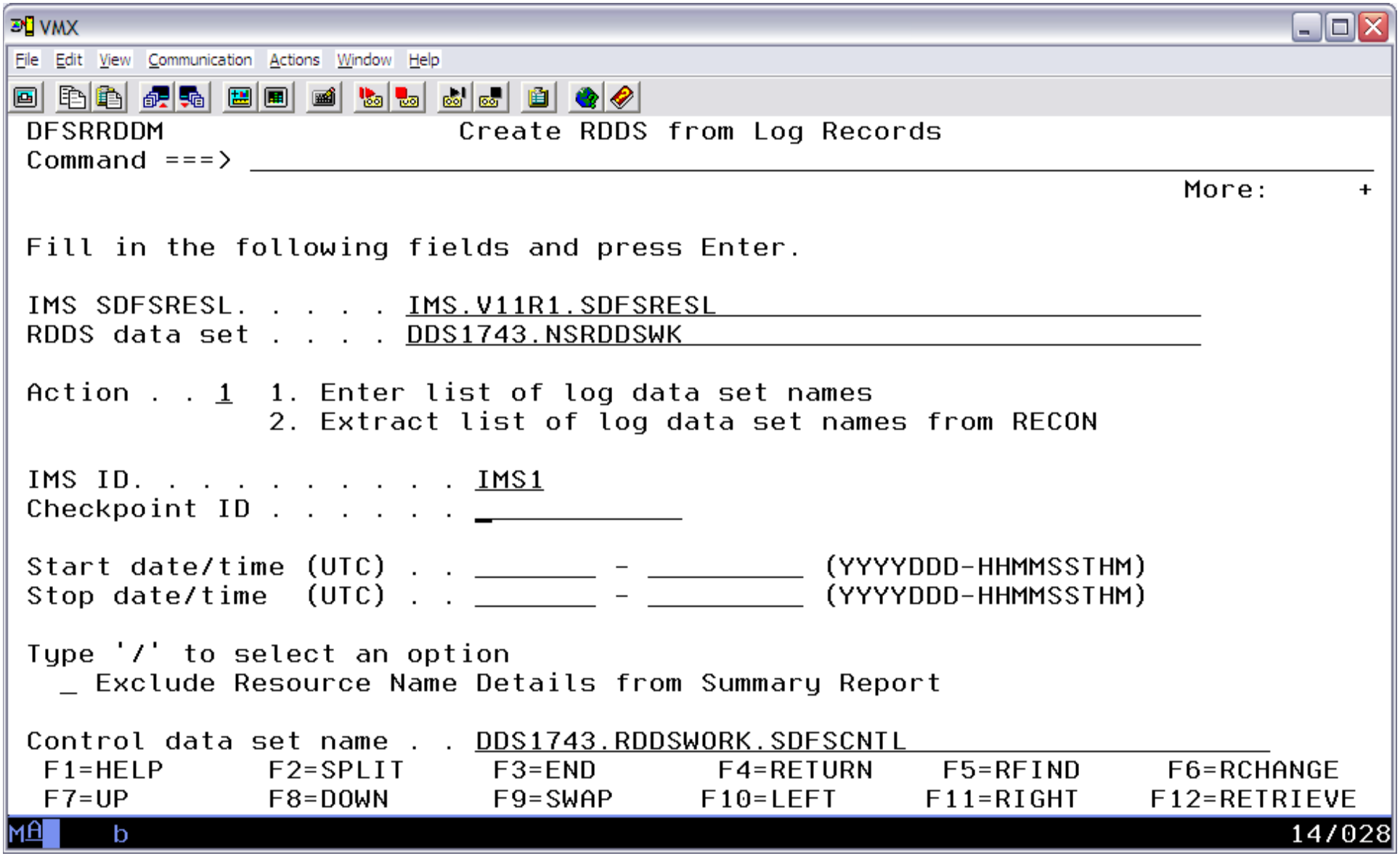
Create RDDS from MODBLKS Utility



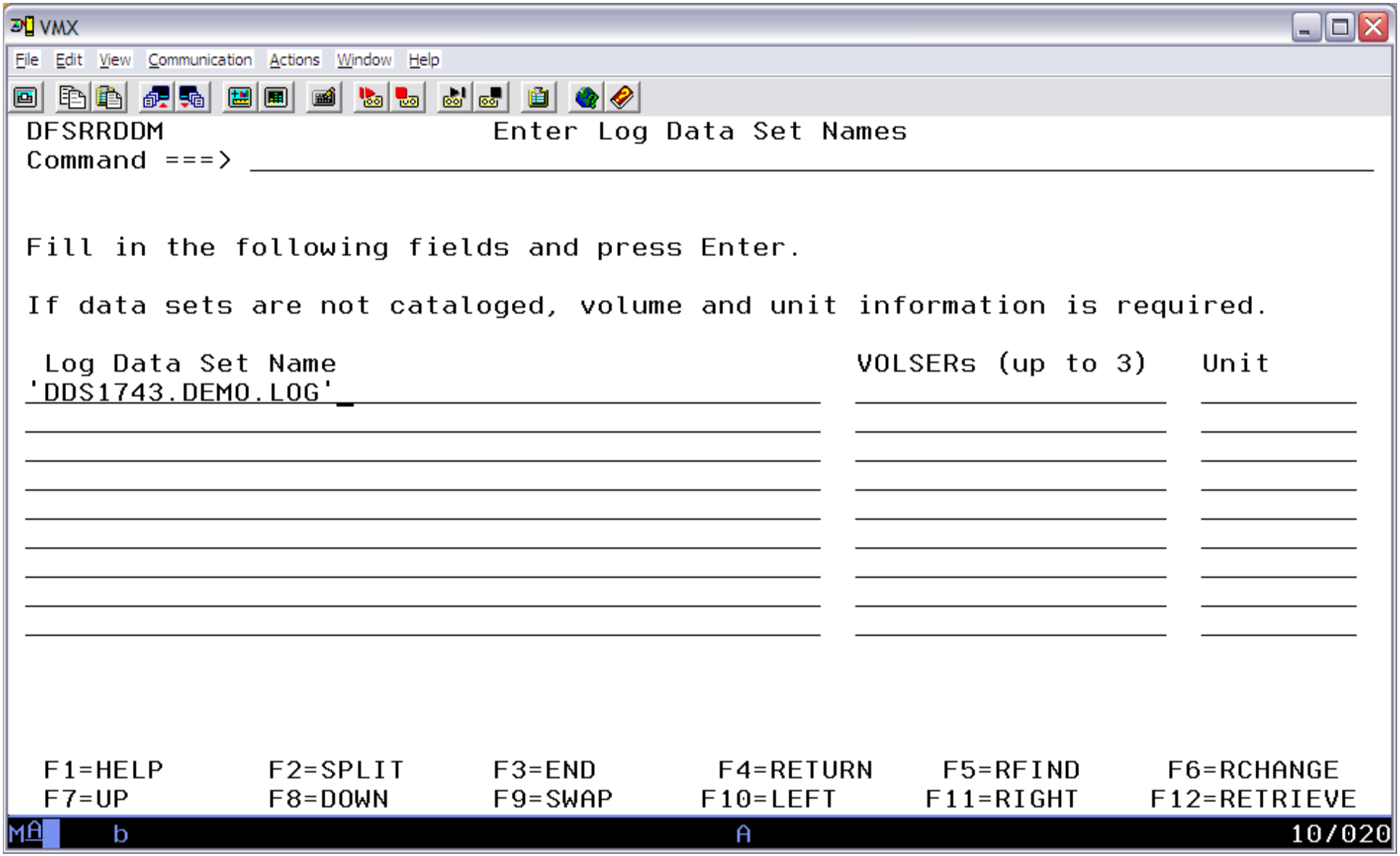
Using DRD Utilities with Log Records

- Log records → RDDS
  - Run Create RDDS from Log Records utility to generate an RDDS using log records as input
    - X'40' checkpoint log records
    - X'22' type-2 command records
  - Examples of when to use
    - If DASD error occurs and need to re-create RDDS
    - Need RDDS in test/sandbox environment to match RDDS in development or production environment

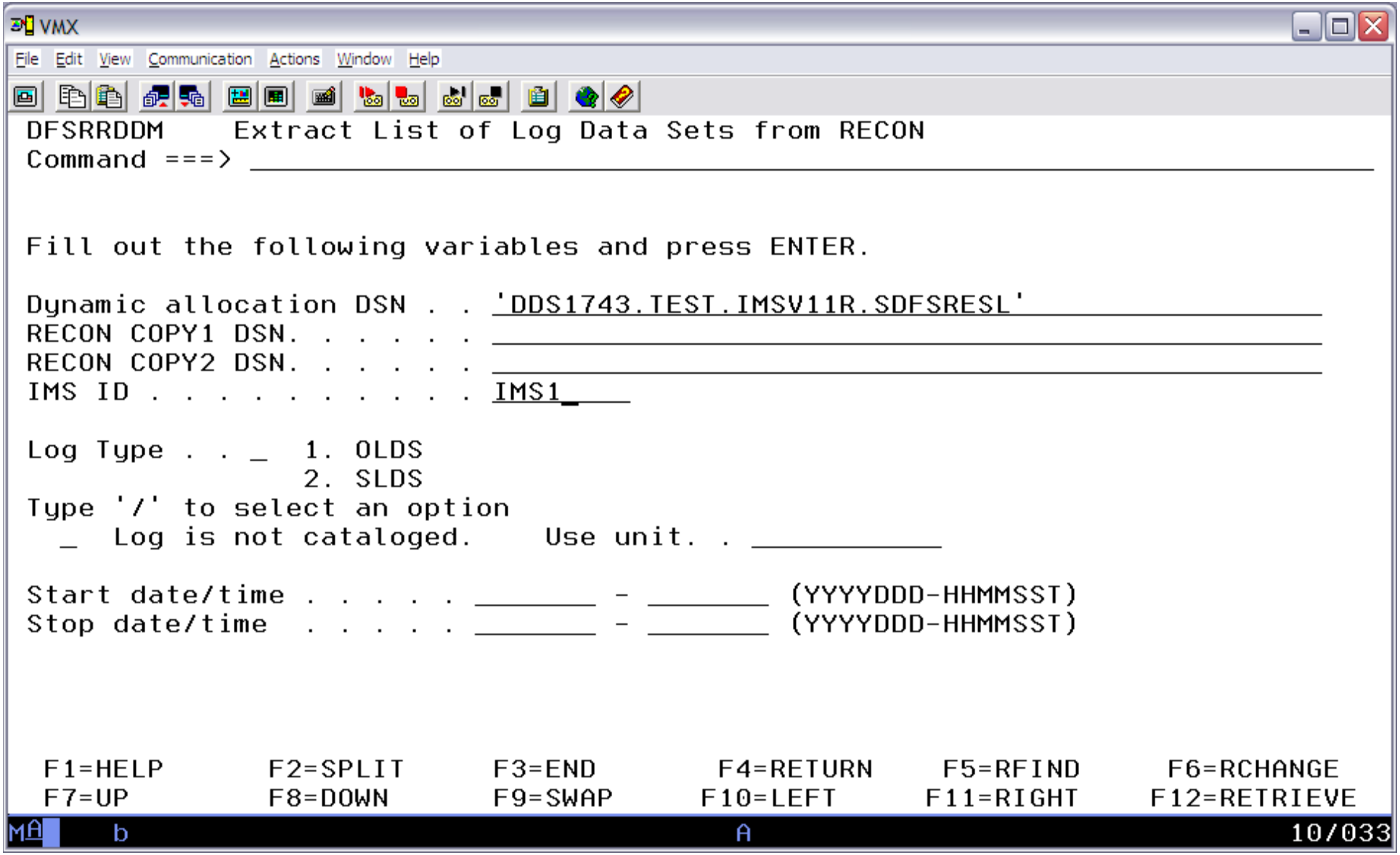
Create RDDS from Log Records Utility



Create RDDS from Log Records Utility

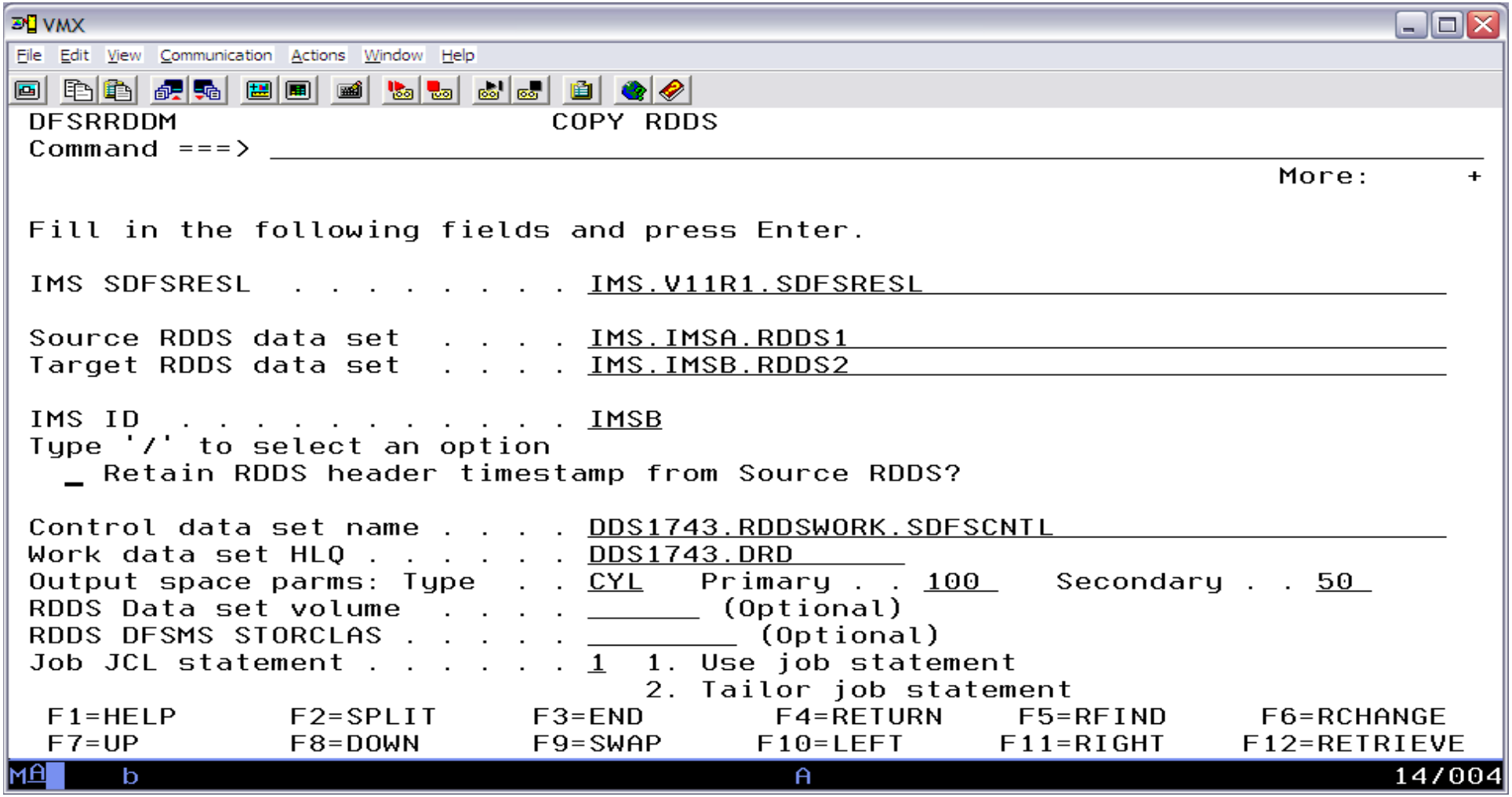


Create RDDS from Log Records Utility



Using DRD Utilities with RDDS

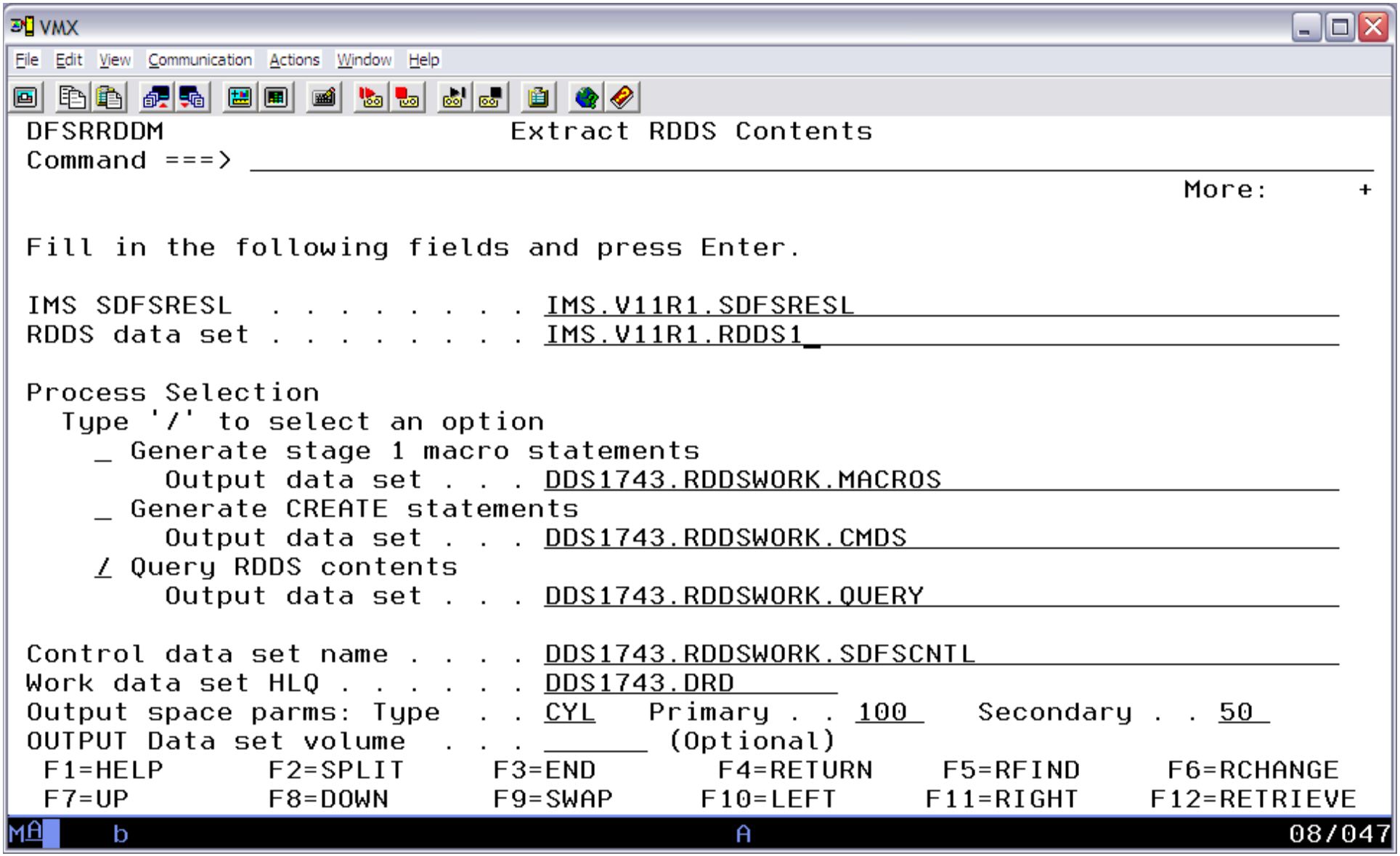
- Need to copy contents of one RDDS to another RDDS
  - Solution: run Copy RDDS utility, specifying source and target RDDSs



Using DRD Utilities for Backup During Migration

- Need to have a matching set of SYSGEN statements in case fallback is needed
  - Solution
    - Run Extract RDDS Contents utility when you want to capture existing definitions
    - Specify “Generate stage 1 macro statements” option
  - Result: a set of SYSGEN macro statements that reflect the IMS system that can be used for SYSGEN of MODBLKS data set
    - Can be reverted to if need to temporarily back out DRD implementation

Transforming RDDS Contents to SYSGEN

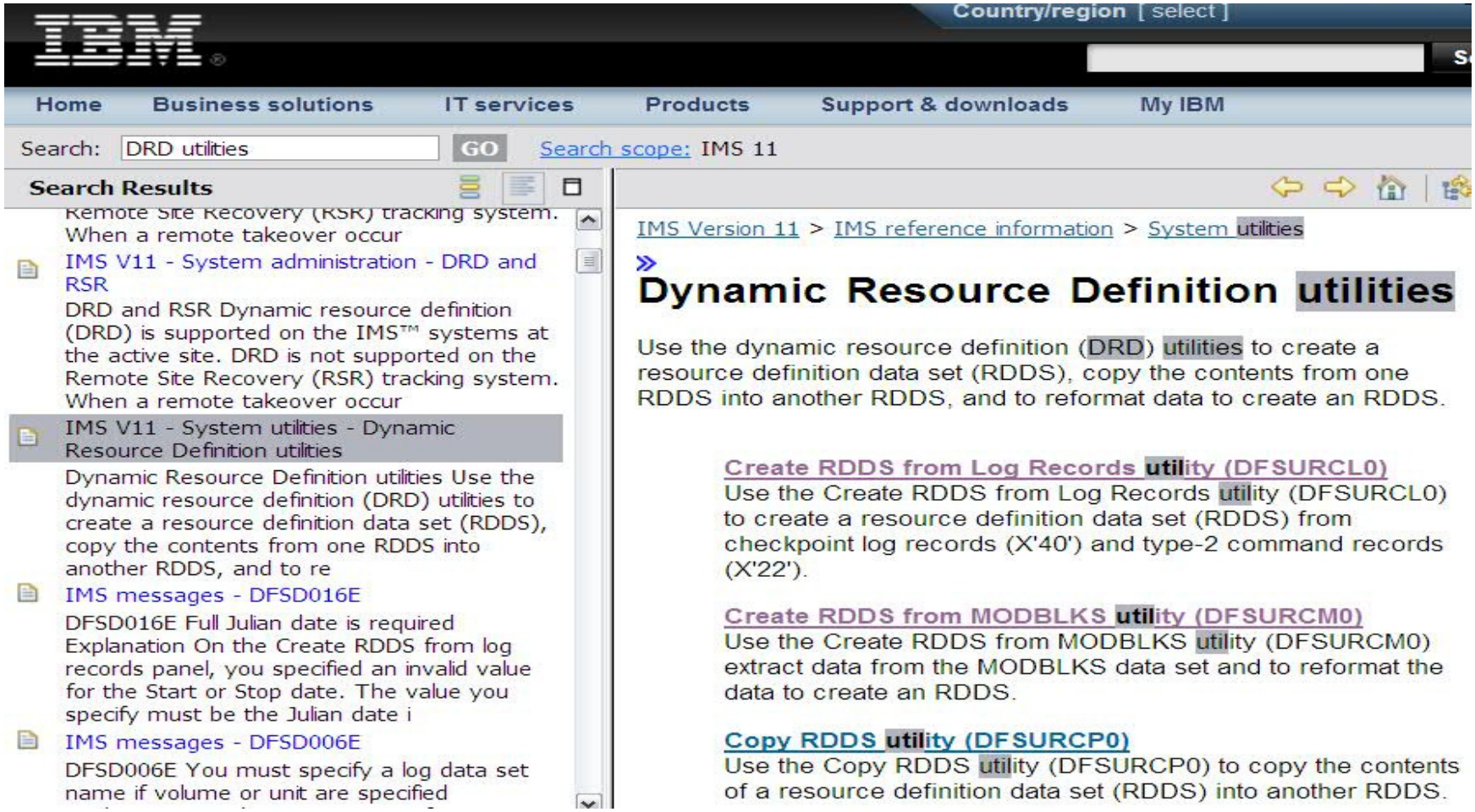


Best Way to Find Guidance for Utilities

- *IMS System Utilities* manual
  - Available in online IMS Information Center
  - F1 field help within Manage Resources interface panels that invoke utilities
  - Speaker notes in this presentation material contain individual utility names + additional detail

DRD Utilities Help

- *System Utilities* manual available in online Info Center



DRD Utilities Help

- 

help within Manage Resources application

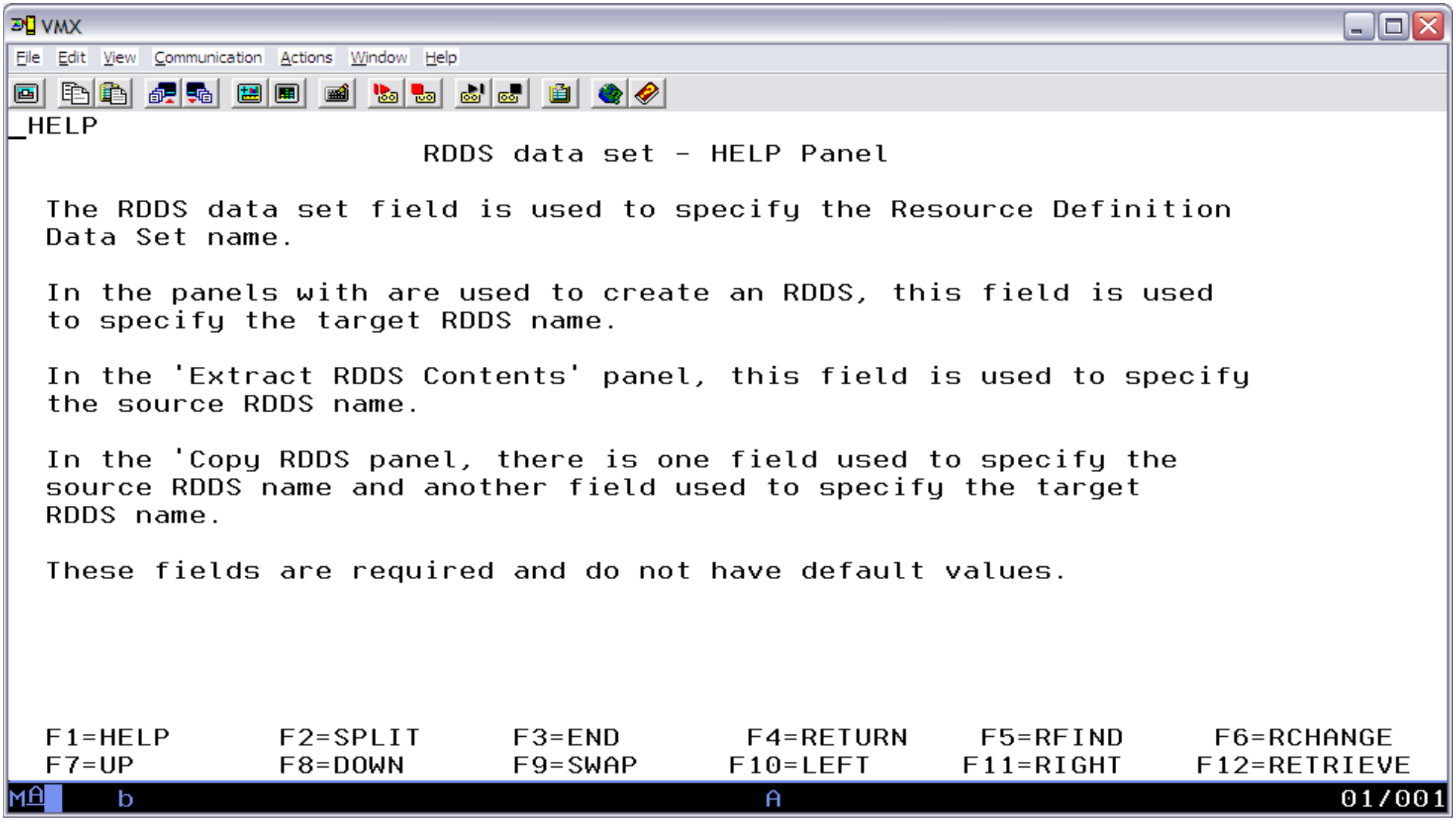
VMX

File Edit View Communication Actions Window Help

```
DFSRRDDM                        Extract RDDS Contents
Command ===> _____________________________________________________________________
                                                                        More:      +

Fill in the following fields and press Enter.

IMS SDFSRESL  . . . . . . . . . IMS.V11R1.SDFSRESL_____________________________
RDDS data set . . . . . . . . . IMS.V11R1.RDDS1_______________________________

Process Selection
  Type '/' to select an option
    / Generate stage 1 macro statements
         Output data set . . . DDS1743.RDDSWORK.MACROS_____________________
    / Generate CREATE statements
         Output data set . . . DDS1743.RDDSWORK.CMDS_______________________
    / Query RDDS contents
         Output data set . . . DDS1743.RDDSWORK.QUERY______________________

Control data set name . . . . DDS1743.RDDSWORK.SDFSCNTL____________________
Work data set HLQ . . . . . . DDS1743.DRD_____
Output space parms: Type  . . CYL   Primary . . 100    Secondary . . 50
OUTPUT Data set volume  . . . _____ (Optional)
 F1=HELP      F2=SPLIT     F3=END       F4=RETURN    F5=RFIND     F6=RCHANGE
 F7=UP        F8=DOWN      F9=SWAP      F10=LEFT     F11=RIGHT    F12=RETRIEVE
MA    b                        A                                   08/032
```

DRD Utilities Help

- Example of F1 help panel for “RDDS data set” field



DRD Usage Best Practices and Avoiding Common Pitfalls

Commands and Procedures

Deleting/Updating Resources with DRD

- Resource cannot be “in use”, for example:
  - Transaction with messages queued
  - Database referenced by a scheduled program

- Recommendation for deleting or updating a resource:
  - QUERY the resources with SHOW(WORK) specified to confirm no work in progress exists for resource
  - Stop resource before attempting to delete or update

Learning Type-2 Commands

- Become familiar with DRD commands
- UPDATE and QUERY commands all have type-1 command equivalents
  - See *Reference Section* at the end of this presentation for a chart that lists type-1 commands with their type-2 DRD command equivalents
  - Sneak Peek:

Task	Type-1 command	Type-2 command
Create or change the limit on the size of application program output segments allowed in message queues for each GU call.	/ASSIGN SEGSZ new_segsize_number TO TRAN tranname	UPDATE TRAN NAME(tranname) SET(SEGSZ(new_segment_size))
Change the class number of a transaction.	/ASSIGN TRAN tranname TO CLS new_class_number	UPDATE TRAN NAME(tranname) SET(CLASS(new_class_number))
Stop updates to a database.	/DBDUMP DB dbname	UPDATE DB NAME(dbname) STOP(UPDATES) OPTION(FEOV)[1]

Creating New Databases

- When adding a new database with DRD, create a corresponding DBD in ACBLIB using online change
  - Traditional (local) OLC
  - Global OLC
  - Member OLC
    - Highest availability, quiesces least amount of members
    - Does not support MSDBs
- Otherwise: newly created database will have a NOTINIT status until this ACBLIB step is completed

Creating New Databases

- Connect other IMS resources to newly created database, such as programs and transactions
- Choose to use DRD for managing MODBLKS resources or online change
  - Coldstart required to switch between the two, which can impact availability

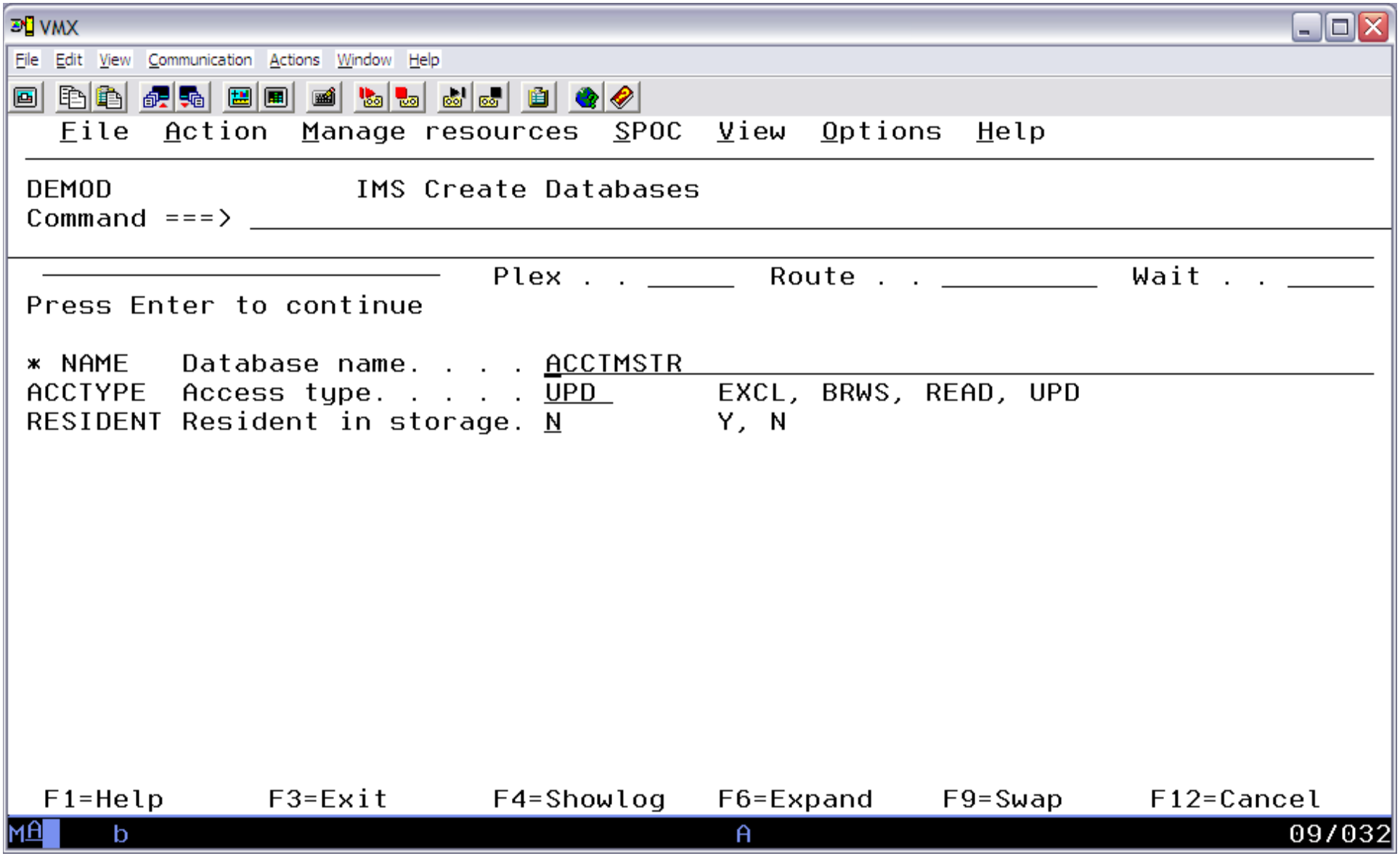
Resources Created using LIKE()

- If updating a resource/descriptor, resources previously created from it will not automatically be updated
- Use Batch SPOC utility to submit UPDATE commands against resources requiring update separately
- Example:
  - CREATE TRANDESC NAME(TDESC1) SET(SERIAL(N))
  - CREATE TRAN NAME(TRAN1, TRAN2) LIKE(TDESC1)
  - UPDATE TRANDESC NAME(TDESC1) SET(SERIAL(Y))
  - UPDATE TRAN NAME(TRAN1,TRAN2) SET(SERIAL(Y))
    - TRAN1, TRAN2, … will still have SERIAL=N so must update separately

DRD User Interface

- Novice users needing to issue DRD commands can use Manage Resources (MR) user interface application
  - Builds DRD commands without requiring knowledge of command syntax
  - Lists available parameter values
    - Includes defaults automatically
    - Shows description of each parameter

Manage Resources Application Example



DRD Usage Best Practices and Avoiding Common Pitfalls

Procedures

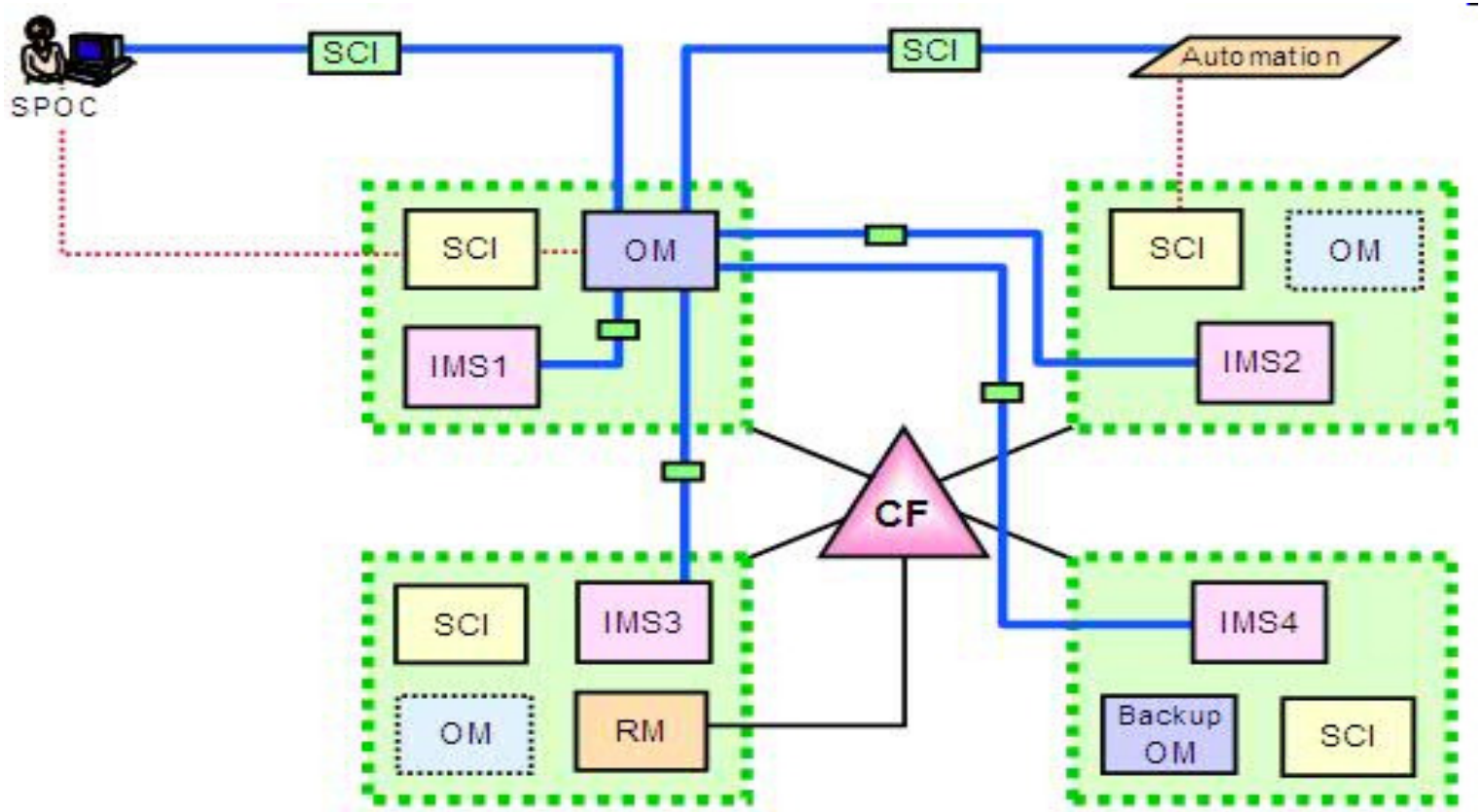
Inadvertently Erasing a Non-System RDDS

- Overwriting non-system RDDS contents with EXPORT command, losing previous resource definitions
- Default EXPORT command parameter is OPTION(OVERWRITE)
  - Solution: Including OPTION(APPEND) in command so that IMS will write to end of non-system RDDS, preserving existing contents

```
TSO SPOC Input
EXPORT DEFN TARGET(RDDS) RDDSDSN(NON.SYS.RDDS1)
  TYPE(ALLDESC) OPTION(APPEND,ALLRSP)

TSO SPOC Output Response
Name      Type      MbrName     CC
PGMTEST   PGMDESC   IMS1         0
TEST      TRANDESC  IMS1         0
```

No Backup OM Address Space

- OM fails, not able to issue DRD or other type-2 commands
  - Solution: Including a backup OM in the IMSplex



Backing Out DRD changes

- Need to “undelete” a resource
  - Before the DELETE command (or series of DELETEs if doing in batch), export definitions to non-system RDDS
  - Delete resources
  - If need to “undo” deletions, IMPORT command can be issued against non-system RDDS
- Need to undo a resource update
  - Before the UPDATE command (or series of UPDATEs if doing in batch), export definitions to non-system RDDS
  - Update resources
  - If need to undo updates, delete updated resources, IMPORT command can be issued against non-system RDDS

Backing Out DRD changes

- Need to undo creating a resource
  - Issue DELETE command against unwanted resources
- Recommendation to not include DELETE in automation, only do ad hoc
- Use IMS tool “IMS Configuration Manager”
  - Uses a type-2 command interface “resource installer” with backout ability if installation errors occur

Timing of DRD Changes

- Planning to make resource changes via DRD but anticipating an IMS shutdown/coldstart
  - Submit changes before shutdown or after coldstart?
    - Attempt changes after coldstart since work in progress less likely then

Batch DRD Updates

- Change management process that requires batch updates
  - Use Batch SPOC utility to submit DRD commands to IMS systems
    - Run in batch, commands submitted via JCL statements
    - Example

```
//SPOCJOB   JOB ,
//MSGCLASS=H,NOTIFY=&SYSUID,USER=&SYSUID
//SPOC      EXEC PGM=CSLUSPOC,
//  PARM=('IMSPLEX=PLEX1,ROUTE=IMS3,WAIT=30')
//STEPLIB   DD DISP=SHR,DSN=IMS.SDFSRESL
//SYSPRINT  DD SYSOUT=*
//SYSIN     DD *
 CREATE TRAN NAME(TRAN1,TRAN2) SET(SERIAL(Y))
 UPDATE TRAN NAME(TRAN3) SET(PARLIM(65535))
/*EOF
```

Track DRD Activity with OM Audit Trail

- Enable OM Audit Trail to track IMSplex activity, including resources being dynamically managed
  - Displays DRD input commands + responses
  - Includes timestamps
  - Example showing an updated database and program…

OM Audit Trail Showing DRD Activity

- Enable OM Audit Trail to track IMSplex activity, including resources being dynamically managed
  - Displays DRD input commands + responses
  - Includes timestamps

```
  File  Action  Manage resources  SPOC  View  Options  Help
PLEX1                          IMSplex Audit Trail
Command ===>

                                    Members . .          Type . .
                                                                More:    -+>
MbrName  Time                      Message
USRT004  2008.149 09:43:47.14 Cmd input . : QRY DB NAME(B*) SHOW(ALL)
USRT004  2008.149 09:43:47.14 Response for: QRY DB NAME(B*) SHOW(ALL)
USRT004  2008.149 09:44:13.42 Cmd input . : UPD DB NAME(BANKTERM) SET(RESIDENT(Y
USRT004  2008.149 09:44:13.42 Response for: UPD DB NAME(BANKTERM) SET(RESIDENT(Y
USRT005  2008.149 09:44:54.83 Cmd input . : QRY MEMBER TYPE(IMS) SHOW(ATTRIB)
USRT005  2008.149 09:44:54.83 Response for: QRY MEMBER TYPE(IMS) SHOW(ATTRIB)
USRT005  2008.149 09:45:02.18 Cmd input . : QRY TRAN SHOW(ALL) STATUS(DYN,IOPREV
USRT005  2008.149 09:45:02.18 Response for: QRY TRAN SHOW(ALL) STATUS(DYN,IOPREV
USRT005  2008.149 09:45:25.23 Cmd input . : QRY DB SHOW(ALL) STATUS(ALLOCF,BACKO
USRT005  2008.149 09:45:25.23 Response for: QRY DB SHOW(ALL) STATUS(ALLOCF,BACKO
USRT001  2008.149 09:46:38.78 Cmd input . : QRY MEMBER TYPE(IMS) SHOW(ATTRIB)
USRT001  2008.149 09:46:38.78 Response for: QRY MEMBER TYPE(IMS) SHOW(ATTRIB)
USRT001  2008.149 09:46:42.76 Cmd input . : QRY PGM SHOW(ALL)
USRT001  2008.149 09:46:42.76 Response for: QRY PGM SHOW(ALL)
USRT001  2008.149 09:47:03.33 Cmd input . : UPD PGM NAME(APOL1)  SET(DOPT(Y))
  F1=Help    F3=Exit    F5=Rfind   F7=Up      F8=Down   F12=Cancel
```

Determining Oldest/Newest RDDS

- Use Extract RDDS Contents utility to generate a query report for each RDDS

```
RDDS HEADER RECORD                                                    +
     HEADER_LENGTH(168) VERSION(1) STATUS(GOOD)                       +
     IMSID(SYS3) IMSTYPE(DBDC) SYSTEM_RDDS?(Y)                        +
     TIMESTAMP(2008.058 21:50:07.695470-UTC)                          +
     data set NAME(USERID.TEST.RDDS2                              )
DB NAME(AUTODB) ACCTYPE(UPD) RESIDENT(N) GLOBAL DMB(0000) +
     LOCAL DMB(0001) MODELNAME() MODELTYPE() TMCR(2007.311 16:18:42.49-UTC)
+
     TMAC() TMUP() TIMP()
DB NAME(AUTODBH) ACCTYPE(UPD) RESIDENT(N) GLOBAL DMB(0000) +
     LOCAL DMB(0002) MODELNAME() MODELTYPE() TMCR(2007.311 16:18:42.49-UTC)
+
     TMAC() TMUP() TIMP()
DB NAME(BANKATMS) ACCTYPE(EXCL) RESIDENT(N) GLOBAL DMB(0000) +
     LOCAL DMB(0003) MODELNAME() MODELTYPE() TMCR(2007.311 16:18:42.49-UTC)
+
TMAC() TMUP() TIMP()
PGM NAME(EMHPSB2) BMPTYPE(N) DOPT(N) FP(E)GPSB(N) +
     RESIDENT(N) SCHDTYPE(PARALLEL) TRANSTAT(N) MODELNAME() +
     MODELTYPE() TMCR(2008.354 22:17:41.80-UTC) TMAC() +
```

Determining Oldest/Newest System RDDS

- Browse each system RDDS using ISPF to view timestamp in header and compare
  - Timestamp written to the system RDDS header during autoexport and included in DFS3371I message
- Automatic import will always read the most current system RDDS
  - In the IMS control region job log, find DFS3395I AUTOMATIC IMPORT STARTED FROM <RDDS name>

DRD Security Considerations

Set RACF Definitions for DRD Commands

Command	Resource Keyword	RACF Access Auth	Resource Name
CREATE	DB	UPDATE	IMS.plxname.CRE.DB
CREATE	DBDESC	UPDATE	IMS.plxname.CRE.DBDESC
CREATE	PGM	UPDATE	IMS.plxname.CRE.PGM
CREATE	PGMDESC	UPDATE	IMS.plxname.CRE.PGMDESC
CREATE	RTC	UPDATE	IMS.plxname.CRE.RTC
CREATE	RTCDESC	UPDATE	IMS.plxname.CRE.RTCDESC
CREATE	TRAN	UPDATE	IMS.plxname.CRE.TRAN
CREATE	TRANDESC	UPDATE	IMS.plxname.CRE.TRANDESC

- Note: the IMSplex name must begin with the characters CSL
- Define in RACF OPERCMDS class

Set RACF Definitions for DRD Commands

Command	Resource Keyword	RACF Access Auth	Resource Name
UPDATE	DB	UPDATE	IMS.plxname.UPD.DB
UPDATE	DBDESC	UPDATE	IMS.plxname.UPD.DBDESC
UPDATE	PGM	UPDATE	IMS.plxname.UPD.PGM
UPDATE	PGMDESC	UPDATE	IMS.plxname.UPD.PGMDESC
UPDATE	RTC	UPDATE	IMS.plxname.UPD.RTC
UPDATE	RTCDESC	UPDATE	IMS.plxname.UPD.RTCDESC
UPDATE	TRAN	UPDATE	IMS.plxname.UPD.TRAN
UPDATE	TRANDESC	UPDATE	IMS.plxname.UPD.TRANDESC

Set RACF Definitions for DRD Commands

Command	Resource Keyword	RACF Access Auth	Resource Name
DELETE	DB	UPDATE	IMS.plxname.DEL.DB
DELETE	DBDESC	UPDATE	IMS.plxname.DEL.DBDESC
DELETE	PGM	UPDATE	IMS.plxname.DEL.PGM
DELETE	PGMDESC	UPDATE	IMS.plxname.DEL.PGMDESC
DELETE	RTC	UPDATE	IMS.plxname.DEL.RTC
DELETE	RTCDESC	UPDATE	IMS.plxname.DEL.RTCDESC
DELETE	TRAN	UPDATE	IMS.plxname.DEL.TRAN
DELETE	TRANDESC	UPDATE	IMS.plxname.DEL.TRANDESC

Set RACF Definitions for DRD Commands

Command	Resource Keyword	RACF Access Auth	Resource Name
IMPORT	DEFN	UPDATE	IMS.plxname.IMP.DEFN
EXPORT	DEFN	UPDATE	IMS.plxname.EXP.DEFN

Set RACF Definitions for DRD Commands

Command	Resource Keyword	RACF Access Auth	Resource Name
QUERY	DB	READ	IMS.plxname.QRY.DB
QUERY	DBDESC	READ	IMS.plxname.QRY.DBDESC
QUERY	PGM	READ	IMS.plxname.QRY.PGM
QUERY	PGMDESC	READ	IMS.plxname.QRY.PGMDESC
QUERY	RTC	READ	IMS.plxname.QRY.RTC
QUERY	RTCDESC	READ	IMS.plxname.QRY.RTCDESC
QUERY	TRAN	READ	IMS.plxname.QRY.TRAN
QUERY	TRANDESC	READ	IMS.plxname.QRY.TRANDESC

Reference Section

Equivalent type-1 and type-2 commands (1) (IMS 11 Command Reference, Vol.1, table 15, p. 46)

Task	Type-1 command	Type-2 command
Change the value for the limit count of a transaction.	/ASSIGN LCT new_lmct_number TO TRAN tranname	UPDATE TRAN NAME (tranname) SET(LCT(new_limit_count))
Change the value for the limit priority of a transaction.	/ASSIGN LPRI new_lpri_number TO TRAN tranname	UPDATE TRAN NAME(tranname) SET(LPRI(new_limit_priority))
Change the value for the normal priority of a transaction.	/ASSIGN NPRI new_npri_number TO TRAN tranname	UPDATE TRAN NAME(tranname) SET(NPRI(new_normal_priority))
Change the value for the parallel processing limit count of a transaction.	/ASSIGN PARLIM new_parlim_number TO TRAN tranname	UPDATE TRAN NAME(tranname) SET(PARLIM(new_parallel_limit))
Change the value for the processing limit count of a transaction.	/ASSIGN PLCT new_plmct_number TO TRAN tranname	UPDATE TRAN NAME(tranname) SET(PLCT(new_processing_limit))
Change the limit on the number of application program output segments allowed in message queues for each GU call.	/ASSIGN SEGNO new_segno_number TO TRAN tranname	UPDATE TRAN NAME(tranname) SET(SEGNO(new_segment_number))

Equivalent type-1 and type-2 commands (2)

Task	Type-1 command	Type-2 command
Create or change the limit on the size of application program output segments allowed in message queues for each GU call.	/ASSIGN SEGSZ new_segsize_number TO TRAN tranname	UPDATE TRAN NAME(tranname) SET(SEGSZ(new_segment_size))
Change the class number of a transaction.	/ASSIGN TRAN tranname TO CLS new_class_number	UPDATE TRAN NAME(tranname) SET(CLASS(new_class_number))
Stop updates to a database.	/DBDUMP DB dbname	UPDATE DB NAME(dbname) STOP(UPDATES) OPTION(FEOV)[1]
Stop the accessing and updating of an area.	/DBRECOVERY AREA areaname	UPDATE AREA NAME(areaname) STOP(ACCESS)
Stop the accessing and updating of all areas and databases of the data group.	/DBRECOVERY DATAGRP datagrpname	UPDATE DATAGRP NAME(datagrpname) STOP(ACCESS)
Stop access to the database and take it offline.	/DBRECOVERY DB dbname	UPDATE DB NAME(dbname) STOP(ACCESS) OPTION(FEOV)[1]
Display information about an area.	/DISPLAY AREA	QUERY AREA
Display the status of a database.	/DISPLAY DB dbname1...dbnamen | ALL	QUERY DB

Equivalent type-1 and type-2 commands (3)

Task	Type-1 command	Type-2 command
Display work in progress for runtime resource definitions that would cause a DELETE, /MODIFY COMMIT, INITIATE OLC PHASE(COMMIT), or UPDATE command to change resource definitions to fail.	/DISPLAY MODIFY ALL	QUERY DB NAME(dbname) SHOW(WORK), QUERY PGM NAME(pgmname) SHOW(WORK), QUERY RTC NAME(rtcode) SHOW(WORK), and QUERY TRAN NAME(tranname) SHOW(WORK).
Display information about a program.	/DISPLAY PGM pgmname	QRY PGM NAME(pgmname) SHOW(ALL) and QUERY PGM NAME(pgmname) SHOW(TRAN)
Display transactions, routing codes and databases associated with a PSB.	/DISPLAY PSB psbname	QUERY DB NAME(dbname) SHOW(PGM), QUERY PGM NAME(pgmname) SHOW(DB), QUERY PGM NAME(pgmname) SHOW(RTC) and QUERY PGM NAME(pgmname) SHOW(TRAN)
Display information about one or more Fast Path routing codes.	/DISPLAY RTC rtcname1...rtcnamen | ALL	QUERY RTC NAME(rtcname1,...rtcnamen | *) SHOW(ALL)

Equivalent type-1 and type-2 commands (4)

Task	Type-1 command	Type-2 command
Display information about databases with the specified status.	/DISPLAY STATUS DB	QUERY DB STATUS(ALLOCF, ALLOCS,BACKOUT,EEQE, LOCK,NOTINIT,NOTOPEN, OFR,OLR,OPEN,RECALL,RECOV, RNL,STOSCHD,STOUPDS
Display all programs that have status and what that status is.	/DISPLAY STATUS PGM	QUERY PGM STATUS(DB-NOTAVL, IOPREV,LOCK,NOTINIT, STOSCHD,TRACE)
Display all Fast Path routing codes that have status and what that status is.	/DISPLAY STATUS RTC	QUERY RTC STATUS(ACTIVE,NOTINIT, NOTSCHD,STOQ)
Display information about transactions with the specified status.	/DISPLAY STATUS TRANSACTION	QUERY TRAN NAME(tranname) STATUS (IOPREV,LCK,QERR,SUSPEND, STOQ,STOSCHD,USTO)
Display information about a transaction.	/DISPLAY TRAN tranname	QUERY TRAN NAME(tranname) SHOW(ALL)
Display all of the transactions.	/DISPLAY TRAN ALL	QUERY TRAN SHOW(ALL)

Equivalent type-1 and type-2 commands (5)

Task	Type-1 command	Type-2 command
Display all of the transactions on the shared queues with a global queue count.	/DISPLAY TRAN tranname QCNT	QUERY TRAN NAME(tranname) SHOW(QCNT)
Stop the use of a database.	/LOCK DB dbname	UPDATE DB NAME(dbname) SET(LOCK(ON))
Lock a program.	/LOCK PGM pgmname	UPDATE PGM NAME(pgmname) SET(LOCK(ON))
Lock a transaction.	/LOCK TRAN tranname	UPDATE TRAN NAME(tranname) SET(LOCK(ON))
Change the transaction so that it is local and runs on the local system.	/MASSIGN TRAN tranname TO LOCAL	UPDATE TRAN NAME(tranname) SET(REMOTE(N))
Change the transaction so that it is remote, and assign it to a specific logical link path.	/MASSIGN TRAN tranname TO MSNAME msname	UPDATE TRAN NAME(tranname) SET(MSNAME(name))

Equivalent type-1 and type-2 commands (6)

Task	Type-1 command	Type-2 command
Stop the scheduling of transactions.	/PSTOP TRAN tranname	UPDATE TRAN(tranname) START(Q) STOP(SCHD)
Stop input messages for a particular transaction code.	/PURGE TRAN tranname	UPDATE TRAN NAME(tranname) START(SCHD) STOP(Q)
Start the area.	/START AREA areaname	UPDATE AREA NAME(areaname) START(ACCESS)
Start the data group.	/START DATAGRP datagrpname	UPDATE DATAGRP NAME(datagrpname) START(ACCESS)
Start a database and change access intent of the database.	/START DB ACCESS	UPDATE DB START(ACCESS) SET(ACCTYPE())
Start a database.	/START DB dbname	UPDATE DB NAME(dbname) START(ACCESS)
Start program scheduling.	/START PGM pgmname	UPDATE PGM NAME(pgmname) START(SCHD)
Start queueing to a Fast Path routing code.	/START RTC rtcname	UPDATE RTC NAME(rtcname) START(Q)

Equivalent type-1 and type-2 commands (7)

Task	Type-1 command	Type-2 command
Start a transaction.	/START TRAN	UPDATE TRAN NAME(tranname) START(Q,SCHD,SUSPEND)
Stop an area.	/STOP AREA areaname	UPDATE AREA NAME(areaname) STOP(SCHD)
Stop a data group.	/STOP DATAGRP datagrpname	UPDATE DATAGRP NAME(datagrpname) STOP(SCHD)
Stop a database.	/STOP DB dbname	UPDATE DB NAME(dbname) STOP(SCHD)
Stop program scheduling.	/STOP PGM pgmname	UPDATE PGM NAME(pgmname) STOP(SCHD)
Stop the queuing and scheduling of messages destined for a transaction.	/STOP TRAN tranname	UPDATE TRAN NAME(tranname) STOP(Q,SCHD)
Start the tracing of a program.	/TRACE SET ON PGM pgmname	UPDATE PGM NAME(pgmname) START(TRACE)
Stop the tracing of a program.	/TRACE SET OFF PGM pgmname	UPDATE PGM NAME(pgmname) STOP(TRACE)

Equivalent type-1 and type-2 commands (8)

Task	Type-1 command	Type-2 command
Stop queueing to a Fast Path routing code.	/STOP RTC rtcname	UPDATE RTC NAME(rtcname) STOP(Q)
Start the tracing of a transaction.	/TRACE SET ON TRAN tranname	UPDATE TRAN NAME(tranname) START(TRA)
Stop the tracing of a transaction.	/TRACE SET OFF TRAN tranname	UPDATE TRAN NAME(tranname) STOP(TRA)
Unlock a database.	/UNLOCK DB dbname	UPDATE DB NAME(dbname) SET(LOCK(OFF))
Unlock a program.	/UNLOCK PGM pgmname	UPDATE PGM NAME(pgmname) SET(LOCK(OFF))
Unlock a transaction.	/UNLOCK TRAN tranname	UPDATE TRAN NAME(tranname) SET(LOCK(OFF))

Summary

- DRD overview
- DRD migration best practices
  - Data set management
  - Autoimport/Autoexport vs. IMPORT/EXPORT commands
  - Recommended parameter settings
- DRD utilities best practices
  - Leveraging utilities to help fold DRD into existing processes
- DRD usage best practices
  - Commands
  - Procedures
- DRD security considerations

Contact Information

Angelique Greenhaw
415-545-2184
greenhaw@us.ibm.com

Questions?