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

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
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 MODBLKDS dataset
  – Autoexport will occur after coldstart complete, populating the system RDDS with the definitions just read in from MODBLKDS 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

Select an application and press Enter.

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

To exit the application, press F3.
Manage Resources Menu

DEMOD

IMS Manage Resources

Command ==> ________________________________

Select an action and press Enter.

* Action . . . . . . _
  1. Create new resources
  2. Delete resources
  3. Query resources
  4. Update resources
  5. Export resources
  6. Import resources
  7. Manage RDDS

F1=Help    F12=Cancel
Manage RDDS Menu

DFS RRDDS
Command ==> 

Select an action and press Enter.

ACTION
1. Create RDDS from Log Records
2. Create RDDS from SYSGEN
3. Create RDDS from MODLKS
4. Extract RDDS Contents
5. Copy RDDS

F1=HELP    F2=SPLIT    F3=END      F4=RETURN    F5=RFIND
F7=UP       F8=DOWN    F9=SWAP     F10=LEFT     F11=RIGHT
F6=RCHANGE  F12=RETRIEVE
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

Fill in the following fields and press Enter.

IMS SDFSRESL . . . . IMS.V11R1.SDFSRESL
RDDS data set . . . . DDS1743.NSRDDSWK

Process . . 1. Perform selections 2-4 below as a single process
2. Create stage 2 JCL
3. Create temporary MODBLKS
4. Process temporary MODBLKS

SYSGEN input DSN . . IMS.V11R1.STAGE1(C)
SYSGEN copy DSN . . IMS.V11R1.STAGE1
Modgen DSN . . . . . . . . SYS1.MODGEN
USERLIB DSN . . . DDS1743.RDDSWORK.USERLIB
Object DSN . . . . . . . . DDS1743.RDDSWORK.OBJDSET
Temp MODBLKS HLQ . . DDS1743.RDDSWORK.TEMPBLKS
IMS HLQ . . . . . . . . . IMS.V11R1

F1=HELP F2=SPLIT F3=END F4=RETURN F5=RFIND F6=RCHANGE
F7=UP F8=DOWN F9=SWAP F10=LEFT F11=RIGHT F12=RETRIEVE
Create RDDS from SYSGEN Panel (PF8)

VMX

DFSRRDDM Create RDDS from SYSGEN Command ==>

End of data

More: -

IMS HLQ . . . . . . IMS.V11R1
ASM parameter . . . . HLASM

IMS ID. . . . . . . IMS1 System type . . . . DBDC

Type '/' to select an option
  _ Large system generation (LGEN)

Control data set name . . DDS1743.RDDSWORK.SDFSCNTL
Work data set HLQ . . . . DDS1743.DRD
Output space parms: Type . . CYL Primary . . 100 Secondary . . 50
RDDDS Data set volume . . . . . (Optional)
RDDDS DFSMS storage class . . . . . (Optional)
Job JCL statement . . . . . . . . 1 1. Use job statement
                          2. Tailor job statement
                          3. Refresh and tailor job statement

F1=HELP F2=SPLIT F3=END F4=RETURN F5=RFIND F6=RCHANGE
F7=UP F8=DOWN F9=SWAP F10=LEFT F11=RIGHT F12=RETRIEVE
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

DFSRRDDM Create RDDS from MODBLKS
Command ==> _____________________________________________________________

Fill in the following fields and press Enter.

IMS SDFSRESL. . . . . . IMS.V11R1.SDFSRESL
RDDS data set . . . . . DDS1743.NSRDDSWK

MODBLKS data set . . . IMS.V11R1.MODBLKS
NUCLEUS data set . . . IMS.V11R1.SDFSRESL

IMS ID. . . . . . . . IMS1
System type . . . . . DBD6C
Suffix . . . . . . . . C

Control data set name . . DDS1743.RDDSWORK.SDFSCNTL
Work data set HLQ . . DDS1743.DRD
Output space parms: Type . . CYL Primary . . 100 Secondary . . 50
RDDS Data set volume . . . _______ (Optional)
RDDS DFSMS STORCLAS. . . _______ (Optional)
Job JCL statement. . . . 1 1. Use job statement
F1=HELP F2=SPLIT F3=END F4=RETURN F5=RFIND F6=RCHANGE
F7=UP F8=DOWN F9=SWAP F10=LEFT F11=RIGHT F12=RETRIEVE
Using DRD Utilities with Log Records

- Log records $\rightarrow$ 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

DFSRRDDM
Command ==>

Fill in the following fields and press Enter.

IMS SDFSRESL. . . . . IMS.V11R1.SDFSRESL
RDDS data set . . . . DDS1743.NSRDDSWK

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

IMS ID. . . . . . . . . . . IMS1
Checkpoint ID . . . . __________

Start date/time (UTC) . . _______ - _______ (YYYYDDD-HHMMSSSTHM)
Stop date/time (UTC) . . _______ - _______ (YYYYDDD-HHMMSSSTHM)

Type '/' to select an option
  _ Exclude Resource Name Details from Summary Report

Control data set name . . DDS1743.RDDSWORK.SDFSCNTL
  F1=HELP  F2=SPLIT  F3=END  F4=RETURN  F5=RFIND  F6=RCHANGE
  F7=UP  F8=DOWN  F9=SWAP  F10=LEFT  F11=RIGHT  F12=RETRIEVE
Create RDDS from Log Records Utility

DFSRRDMM Command ===> _________________________________

Enter Log Data Set Names

Fill in the following fields and press Enter.

If data sets are not cataloged, volume and unit information is required.

Log Data Set Name
'DDS1743.DEML0G'

VOLSERs (up to 3)
Unit

F1=HELP
F7=UP
F2=SPLIT
F8=DOWN
F3=END
F9=SWAP
F4=RETURN
F10=LEFT
F5=RFIND
F11=RIGHT
F6=RCHANGE
F12=RETRIEVE
Create RDDS from Log Records Utility

DFSRRDDM Extract List of Log Data Sets from RECON
Command ==> ____________________________

Fill out the following variables and press ENTER.

Dynamic allocation DSN . . 'DDS1743.TEST.IMSV11R.SDFSRESI'
RECON COPY1 DSN. . . . ____________________________
RECON COPY2 DSN. . . . ____________________________
IMS ID . . . . . . . . . . IMS1___

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

Start date/time . . . ________ - ________ (YYYYDDD-HHMMSS)
Stop date/time . . . ________ - ________ (YYYYDDD-HHMMSS)

F1=HELP         F2=SPLIT         F3=END         F4=RETURN
F7=UP           F8=DOWM          F9=SWAP        F10=LEFT
F11=RIGHT       F12=RETRIEVE
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

DFSRDDM Command ==> ________________________________

Extract RDDS Contents

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 parm: 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
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

- **F1** help within Manage Resources application
Example of F1 help panel for “RDDS data set” field

The RDDS data set field is used to specify the Resource Definition Data Set name.

In the panels with are used to create an RDDS, this field is used to specify the target RDDS name.

In the 'Extract RDDS Contents' panel, this field is used to specify the source RDDS name.

In the 'Copy RDDS panel, there is one field used to specify the source RDDS name and another field used to specify the target RDDS name.

These fields are required and do not have default values.
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:

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

DEMOD Command ==> IMS Create Databases

Plex . . _____ Route . . _______ Wait . . ___
Press Enter to continue

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

F1=Help F3=Exit F4=Showlog F6=Expand F9=Swap F12=Cancel
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**
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>MbrName</th>
<th>CC</th>
</tr>
</thead>
<tbody>
<tr>
<td>PGMTEST</td>
<td>PGMDESC</td>
<td>IMS1</td>
<td>0</td>
</tr>
<tr>
<td>TEST</td>
<td>TRANDESC</td>
<td>IMS1</td>
<td>0</td>
</tr>
</tbody>
</table>
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

```plaintext
//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

```
<table>
<thead>
<tr>
<th>MbrName</th>
<th>Time</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>USRT004</td>
<td>2008.149 09:43:47.14</td>
<td>Cmd input: QRY DB NAME(B.*) SHOW(ALL)</td>
</tr>
<tr>
<td>USRT004</td>
<td>2008.149 09:43:47.14</td>
<td>Response for: QRY DB NAME(B.*) SHOW(ALL)</td>
</tr>
<tr>
<td>USRT004</td>
<td>2008.149 09:44:13.42</td>
<td>Cmd input: UPD DB NAME(BANKTERM) SET(RESIDENT(Y</td>
</tr>
<tr>
<td>USRT004</td>
<td>2008.149 09:44:13.42</td>
<td>Response for: UPD DB NAME(BANKTERM) SET(RESIDENT(Y</td>
</tr>
<tr>
<td>USRT005</td>
<td>2008.149 09:44:54.83</td>
<td>Cmd input: QRY MEMBER TYPE(IMS) SHOW(ATTRIB)</td>
</tr>
<tr>
<td>USRT005</td>
<td>2008.149 09:44:54.83</td>
<td>Response for: QRY MEMBER TYPE(IMS) SHOW(ATTRIB)</td>
</tr>
<tr>
<td>USRT005</td>
<td>2008.149 09:45:02.18</td>
<td>Cmd input: QRY TRAN SHOW(ALL) STATUS(DYN,IOPREV</td>
</tr>
<tr>
<td>USRT005</td>
<td>2008.149 09:45:02.18</td>
<td>Response for: QRY TRAN SHOW(ALL) STATUS(DYN,IOPREV</td>
</tr>
<tr>
<td>USRT005</td>
<td>2008.149 09:45:25.23</td>
<td>Cmd input: QRY DB SHOW(ALL) STATUS(ALLOCF,BACKO</td>
</tr>
<tr>
<td>USRT005</td>
<td>2008.149 09:45:25.23</td>
<td>Response for: QRY DB SHOW(ALL) STATUS(ALLOCF,BACKO</td>
</tr>
<tr>
<td>USRT001</td>
<td>2008.149 09:46:38.78</td>
<td>Cmd input: QRY MEMBER TYPE(IMS) SHOW(ATTRIB)</td>
</tr>
<tr>
<td>USRT001</td>
<td>2008.149 09:46:38.78</td>
<td>Response for: QRY MEMBER TYPE(IMS) SHOW(ATTRIB)</td>
</tr>
<tr>
<td>USRT001</td>
<td>2008.149 09:46:42.76</td>
<td>Cmd input: QRY PGM SHOW(ALL)</td>
</tr>
<tr>
<td>USRT001</td>
<td>2008.149 09:46:42.76</td>
<td>Response for: QRY PGM SHOW(ALL)</td>
</tr>
<tr>
<td>USRT001</td>
<td>2008.149 09:47:03.33</td>
<td>Cmd input: UPD PGM NAME(APOL1) SET(DOPT(Y))</td>
</tr>
</tbody>
</table>
```

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) +
  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

<table>
<thead>
<tr>
<th>Command</th>
<th>Resource Keyword</th>
<th>RACF Access Auth</th>
<th>Resource Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>CREATE</td>
<td>DB</td>
<td>UPDATE</td>
<td>IMS.plxname.CRE.DB</td>
</tr>
<tr>
<td>CREATE</td>
<td>DBDESC</td>
<td>UPDATE</td>
<td>IMS.plxname.CRE.DBDESC</td>
</tr>
<tr>
<td>CREATE</td>
<td>PGM</td>
<td>UPDATE</td>
<td>IMS.plxname.CRE.PGM</td>
</tr>
<tr>
<td>CREATE</td>
<td>PGMDESC</td>
<td>UPDATE</td>
<td>IMS.plxname.CRE.PGMDESC</td>
</tr>
<tr>
<td>CREATE</td>
<td>RTC</td>
<td>UPDATE</td>
<td>IMS.plxname.CRE.RTC</td>
</tr>
<tr>
<td>CREATE</td>
<td>RTCDESC</td>
<td>UPDATE</td>
<td>IMS.plxname.CRE.RTCDESC</td>
</tr>
<tr>
<td>CREATE</td>
<td>TRAN</td>
<td>UPDATE</td>
<td>IMS.plxname.CRE.TRAN</td>
</tr>
<tr>
<td>CREATE</td>
<td>TRANDESC</td>
<td>UPDATE</td>
<td>IMS.plxname.CRE.TRANDESC</td>
</tr>
</tbody>
</table>

- Note: the IMSplex name must begin with the characters CSL
- Define in RACF OPERCMDS class
## Set RACF Definitions for DRD Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Resource Keyword</th>
<th>RACF Access Auth</th>
<th>Resource Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>UPDATE</td>
<td>DB</td>
<td>UPDATE</td>
<td>IMS.plxname.UPD.DB</td>
</tr>
<tr>
<td>UPDATE</td>
<td>DBDESC</td>
<td>UPDATE</td>
<td>IMS.plxname.UPD.DBDESC</td>
</tr>
<tr>
<td>UPDATE</td>
<td>PGM</td>
<td>UPDATE</td>
<td>IMS.plxname.UPD.PGM</td>
</tr>
<tr>
<td>UPDATE</td>
<td>PGMDESC</td>
<td>UPDATE</td>
<td>IMS.plxname.UPD.PGMDESC</td>
</tr>
<tr>
<td>UPDATE</td>
<td>RTC</td>
<td>UPDATE</td>
<td>IMS.plxname.UPD.RTC</td>
</tr>
<tr>
<td>UPDATE</td>
<td>RTCDESC</td>
<td>UPDATE</td>
<td>IMS.plxname.UPD.RTCDESC</td>
</tr>
<tr>
<td>UPDATE</td>
<td>TRAN</td>
<td>UPDATE</td>
<td>IMS.plxname.UPD.TRAN</td>
</tr>
<tr>
<td>UPDATE</td>
<td>TRANDESC</td>
<td>UPDATE</td>
<td>IMS.plxname.UPD.TRANDESC</td>
</tr>
</tbody>
</table>
## Set RACF Definitions for DRD Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Resource Keyword</th>
<th>RACF Access Auth</th>
<th>Resource Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>DELETE</td>
<td>DB</td>
<td>UPDATE</td>
<td>IMS.plxname.DEL.DB</td>
</tr>
<tr>
<td>DELETE</td>
<td>DBDESC</td>
<td>UPDATE</td>
<td>IMS.plxname.DEL.DBDESC</td>
</tr>
<tr>
<td>DELETE</td>
<td>PGM</td>
<td>UPDATE</td>
<td>IMS.plxname.DEL.PGM</td>
</tr>
<tr>
<td>DELETE</td>
<td>PGMDESC</td>
<td>UPDATE</td>
<td>IMS.plxname.DEL.PGMDESC</td>
</tr>
<tr>
<td>DELETE</td>
<td>RTC</td>
<td>UPDATE</td>
<td>IMS.plxname.DEL_RTC</td>
</tr>
<tr>
<td>DELETE</td>
<td>RTCDESC</td>
<td>UPDATE</td>
<td>IMS.plxname.DEL_RTCDESC</td>
</tr>
<tr>
<td>DELETE</td>
<td>TRAN</td>
<td>UPDATE</td>
<td>IMS.plxname.DEL.TRAN</td>
</tr>
<tr>
<td>DELETE</td>
<td>TRANDESC</td>
<td>UPDATE</td>
<td>IMS.plxname.DEL.TRANDESC</td>
</tr>
</tbody>
</table>
## Set RACF Definitions for DRD Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Resource Keyword</th>
<th>RACF Access Auth</th>
<th>Resource Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMPORT</td>
<td>DEFN</td>
<td>UPDATE</td>
<td>IMS.plxname.IMP.DEFN</td>
</tr>
<tr>
<td>EXPORT</td>
<td>DEFN</td>
<td>UPDATE</td>
<td>IMS.plxname.EXP.DEFN</td>
</tr>
</tbody>
</table>
## Set RACF Definitions for DRD Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Resource Keyword</th>
<th>RACF Access Auth</th>
<th>Resource Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>QUERY</td>
<td>DB</td>
<td>READ</td>
<td>IMS.plxname.QRY.DB</td>
</tr>
<tr>
<td>QUERY</td>
<td>DBDESC</td>
<td>READ</td>
<td>IMS.plxname.QRY.DBDESC</td>
</tr>
<tr>
<td>QUERY</td>
<td>PGM</td>
<td>READ</td>
<td>IMS.plxname.QRY.PGM</td>
</tr>
<tr>
<td>QUERY</td>
<td>PGMDESC</td>
<td>READ</td>
<td>IMS.plxname.QRY.PGMDESC</td>
</tr>
<tr>
<td>QUERY</td>
<td>RTC</td>
<td>READ</td>
<td>IMS.plxname.QRY.RTC</td>
</tr>
<tr>
<td>QUERY</td>
<td>RTCDESC</td>
<td>READ</td>
<td>IMS.plxname.QRY.RTCDESC</td>
</tr>
<tr>
<td>QUERY</td>
<td>TRAN</td>
<td>READ</td>
<td>IMS.plxname.QRY.TRAN</td>
</tr>
<tr>
<td>QUERY</td>
<td>TRANDESC</td>
<td>READ</td>
<td>IMS.plxname.QRY.TRANDESC</td>
</tr>
</tbody>
</table>
Reference Section
<table>
<thead>
<tr>
<th>Task</th>
<th>Type-1 command</th>
<th>Type-2 command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Change the value for the limit count of a transaction.</td>
<td>/ASSIGN LCT new_lmct_number TO TRAN tranname</td>
<td>UPDATE TRAN NAME (tranname) SET(LCT(new_limit_count))</td>
</tr>
<tr>
<td>Change the value for the limit priority of a transaction.</td>
<td>/ASSIGN LPRI new_lpri_number TO TRAN tranname</td>
<td>UPDATE TRAN NAME(tranname) SET(LPRI(new_limit_priority))</td>
</tr>
<tr>
<td>Change the value for the normal priority of a transaction.</td>
<td>/ASSIGN NPRI new_npri_number TO TRAN tranname</td>
<td>UPDATE TRAN NAME(tranname) SET(NPRI(new_normal_priority))</td>
</tr>
<tr>
<td>Change the value for the parallel processing limit count of a transaction.</td>
<td>/ASSIGN PARLIM new_parlim_number TO TRAN tranname</td>
<td>UPDATE TRAN NAME(tranname) SET(PARLIM(new_parallel_limit))</td>
</tr>
<tr>
<td>Change the value for the processing limit count of a transaction.</td>
<td>/ASSIGN PLCT new_plmct_number TO TRAN tranname</td>
<td>UPDATE TRAN NAME(tranname) SET(PLCT(new_processing_limit))</td>
</tr>
<tr>
<td>Change the limit on the number of application program output segments allowed in message queues for each GU call.</td>
<td>/ASSIGN SEGNO new_segno_number TO TRAN tranname</td>
<td>UPDATE TRAN NAME(tranname) SET(SEGNO(new_segment_number))</td>
</tr>
</tbody>
</table>
### Equivalent type-1 and type-2 commands (2)

<table>
<thead>
<tr>
<th>Task</th>
<th>Type-1 command</th>
<th>Type-2 command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create or change the limit on the size of application program output segments allowed in message queues for each GU call.</td>
<td><code>/ASSIGN SEGSZ new_segsize_number TO TRAN tranname</code></td>
<td><code>UPDATE TRAN NAME(tranname) SET(SEGSZ(new_segment_size))</code></td>
</tr>
<tr>
<td>Change the class number of a transaction.</td>
<td><code>/ASSIGN TRAN tranname TO CLS new_class_number</code></td>
<td><code>UPDATE TRAN NAME(tranname) SET(CLASS(new_class_number))</code></td>
</tr>
<tr>
<td>Stop updates to a database.</td>
<td><code>/DBDUMP DB dbname</code></td>
<td><code>UPDATE DB NAME(dbname) STOP(UPDATES) OPTION(FEOV)</code></td>
</tr>
<tr>
<td>Stop the accessing and updating of an area.</td>
<td><code>/DBRECOVERY AREA areaname</code></td>
<td><code>UPDATE AREA NAME(areaname) STOP(ACCESS)</code></td>
</tr>
<tr>
<td>Stop the accessing and updating of all areas and databases of the data group.</td>
<td><code>/DBRECOVERY DATAGRP datagrpname</code></td>
<td><code>UPDATE DATAGRP NAME(detagrpname) STOP(ACCESS)</code></td>
</tr>
<tr>
<td>Stop access to the database and take it offline.</td>
<td><code>/DBRECOVERY DB dbname</code></td>
<td><code>UPDATE DB NAME(dbname) STOP(ACCESS) OPTION(FEOV)</code></td>
</tr>
<tr>
<td>Display information about an area.</td>
<td><code>/DISPLAY AREA</code></td>
<td><code>QUERY AREA</code></td>
</tr>
<tr>
<td>Display the status of a database.</td>
<td>`/DISPLAY DB dbname...dbname</td>
<td>ALL`</td>
</tr>
<tr>
<td>Task</td>
<td>Type-1 command</td>
<td>Type-2 command</td>
</tr>
<tr>
<td>------</td>
<td>---------------</td>
<td>----------------</td>
</tr>
<tr>
<td>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.</td>
<td>/DISPLAY MODIFY ALL</td>
<td>QUERY DB NAME(dbname) SHOW(WORK), QUERY PGM NAME(pgmname) SHOW(WORK), QUERY RTC NAME(retcode) SHOW(WORK), and QUERY TRAN NAME(tranname) SHOW(WORK).</td>
</tr>
<tr>
<td>Display information about a program.</td>
<td>/DISPLAY PGM pgmname</td>
<td>QRY PGM NAME(pgmname) SHOW(ALL) and QUERY PGM NAME(pgmname) SHOW(TRAN)</td>
</tr>
<tr>
<td>Display transactions, routing codes and databases associated with a PSB.</td>
<td>/DISPLAY PSB psbname</td>
<td>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)</td>
</tr>
<tr>
<td>Display information about one or more Fast Path routing codes.</td>
<td>/DISPLAY RTC rtcnamel...rtcnamen</td>
<td>ALL</td>
</tr>
</tbody>
</table>
### Equivalent type-1 and type-2 commands (4)

<table>
<thead>
<tr>
<th>Task</th>
<th>Type-1 command</th>
<th>Type-2 command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display information about databases with the specified status.</td>
<td><code>/DISPLAY STATUS DB</code></td>
<td><code>QUERY DB STATUS(ALLOC,F, ALLOCS,BACKOUT,EEQF, LOCK,NOTINIT,NOTOPEN, OFR,OLR,OPEN,RECALL,RECOV, RNL,STOSCHD,STOUPDS)</code></td>
</tr>
<tr>
<td>Display all programs that have status and what that status is.</td>
<td><code>/DISPLAY STATUS PGM</code></td>
<td><code>QUERY PGM STATUS(DB-NOTAVL, IOPREV,LOCK,NOTINIT, STOSCHD,TRACE)</code></td>
</tr>
<tr>
<td>Display all Fast Path routing codes that have status and what that status is.</td>
<td><code>/DISPLAY STATUS RTC</code></td>
<td><code>QUERY RTC STATUS(ACTIVE,NOTINIT, NOTSCHD,STOQ)</code></td>
</tr>
<tr>
<td>Display information about transactions with the specified status.</td>
<td><code>/DISPLAY STATUS TRANSACTION</code></td>
<td><code>QUERY TRAN NAME(trannname) STATUS (IOPREV,LCK,QERR,SUSPEND, STOQ,STOSCHD,USTO)</code></td>
</tr>
<tr>
<td>Display information about a transaction.</td>
<td><code>/DISPLAY TRAN trannname</code></td>
<td><code>QUERY TRAN NAME(trannname) SHOW(ALL)</code></td>
</tr>
<tr>
<td>Display all of the transactions.</td>
<td><code>/DISPLAY TRAN ALL</code></td>
<td><code>QUERY TRAN SHOW(ALL)</code></td>
</tr>
<tr>
<td>Task</td>
<td>Type-1 command</td>
<td>Type-2 command</td>
</tr>
<tr>
<td>------</td>
<td>----------------</td>
<td>----------------</td>
</tr>
<tr>
<td>Display all of the transactions on the shared queues with a global queue count.</td>
<td><code>/DISPLAY TRAN tranname QCNT</code></td>
<td><code>QUERY TRAN NAME(tranname) SHOW(QCNT)</code></td>
</tr>
<tr>
<td>Stop the use of a database.</td>
<td><code>/LOCK DB dbname</code></td>
<td><code>UPDATE DB NAME(dbname) SET(LOCK(ON))</code></td>
</tr>
<tr>
<td>Lock a program.</td>
<td><code>/LOCK PGM pgmname</code></td>
<td><code>UPDATE PGM NAME(pgmname) SET(LOCK(ON))</code></td>
</tr>
<tr>
<td>Lock a transaction.</td>
<td><code>/LOCK TRAN tranname</code></td>
<td><code>UPDATE TRAN NAME(tranname) SET(LOCK(ON))</code></td>
</tr>
<tr>
<td>Change the transaction so that it is local and runs on the local system.</td>
<td><code>/MASSIGN TRAN tranname TO LOCAL</code></td>
<td><code>UPDATE TRAN NAME(tranname) SET(REMOTE(N))</code></td>
</tr>
<tr>
<td>Change the transaction so that it is remote, and assign it to a specific logical link path.</td>
<td><code>/MASSIGN TRAN tranname TO MSNAME msname</code></td>
<td><code>UPDATE TRAN NAME(tranname) SET(MSNAME(name))</code></td>
</tr>
</tbody>
</table>
# Equivalent type-1 and type-2 commands (6)

<table>
<thead>
<tr>
<th>Task</th>
<th>Type-1 command</th>
<th>Type-2 command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stop the scheduling of transactions.</td>
<td>/PSTOP TRAN tranname</td>
<td>UPDATE TRAN(tranname) START(Q) STOP(SCHD)</td>
</tr>
<tr>
<td>Stop input messages for a particular transaction code.</td>
<td>/PURGE TRAN tranname</td>
<td>UPDATE TRAN NAME(tranname) START(SCHD) STOP(Q)</td>
</tr>
<tr>
<td>Start the area.</td>
<td>/START AREA areaname</td>
<td>UPDATE AREA NAME(areaname) START(ACCESS)</td>
</tr>
<tr>
<td>Start the data group.</td>
<td>/START DATAGRP datagrpname</td>
<td>UPDATE DATAGRP NAME(datagrpname) START(ACCESS)</td>
</tr>
<tr>
<td>Start a database and change access intent of the database.</td>
<td>/START DB ACCESS</td>
<td>UPDATE DB START(ACCESS) SET(ACCTYPE())</td>
</tr>
<tr>
<td>Start a database.</td>
<td>/START DB dbname</td>
<td>UPDATE DB NAME(dbname) START(ACCESS)</td>
</tr>
<tr>
<td>Start program scheduling.</td>
<td>/START PGM pgmname</td>
<td>UPDATE PGM NAME(pgmname) START(SCHD)</td>
</tr>
<tr>
<td>Start queueing to a Fast Path routing code.</td>
<td>/START RTC rtcname</td>
<td>UPDATE RTC NAME(rtcname) START(Q)</td>
</tr>
</tbody>
</table>
## Equivalent type-1 and type-2 commands (7)

<table>
<thead>
<tr>
<th>Task</th>
<th>Type-1 command</th>
<th>Type-2 command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start a transaction.</td>
<td>/START TRAN</td>
<td>UPDATE TRAN NAME(trannname) START(Q,SCHD,SUSPEND)</td>
</tr>
<tr>
<td>Stop an area.</td>
<td>/STOP AREA areaname</td>
<td>UPDATE AREA NAME(areaname) STOP(SCHD)</td>
</tr>
<tr>
<td>Stop a data group.</td>
<td>/STOP DATAGRP datagrpname</td>
<td>UPDATE DATAGRP NAME(datagrpname) STOP(SCHD)</td>
</tr>
<tr>
<td>Stop a database.</td>
<td>/STOP DB dbname</td>
<td>UPDATE DB NAME(dbname) STOP(SCHD)</td>
</tr>
<tr>
<td>Stop program scheduling.</td>
<td>/STOP PGM pgmname</td>
<td>UPDATE PGM NAME(pgmname) STOP(SCHD)</td>
</tr>
<tr>
<td>Stop the queuing and scheduling of messages destined for a transaction.</td>
<td>/STOP TRAN tranname</td>
<td>UPDATE TRAN NAME(tranname) STOP(Q,SCHD)</td>
</tr>
<tr>
<td>Start the tracing of a program.</td>
<td>/TRACE SET ON PGM pgmname</td>
<td>UPDATE PGM NAME(pgmname) START(TRACE)</td>
</tr>
<tr>
<td>Stop the tracing of a program.</td>
<td>/TRACE SET OFF PGM pgmname</td>
<td>UPDATE PGM NAME(pgmname) STOP(TRACE)</td>
</tr>
</tbody>
</table>
## Equivalent type-1 and type-2 commands (8)

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