The Junior Woodchuck's Guide to Repairing Your RACFVM Database

A handy guide to performing emergency ops on your security policy

Brian W. Hugenbruch, CISSP
IBM z Systems Virtualization and Cloud Security
bwhugen@us.ibm.com *@Bwhugen*

v.01h – Last updated 16 June 2017

#vmworkshop #zVM #IBMz

Disclaimer

The information contained in this document has not been submitted to any formal IBM test and is distributed on an "AS IS" basis without any warranty either express or implied. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

In this document, any references made to an IBM licensed program are not intended to state or imply that only IBM's licensed program may be used; any functionally equivalent program may be used instead.

Any performance data contained in this document was determined in a controlled environment and, therefore, the results which may be obtained in other operating environments may vary significantly. Users of this document should verify the applicable data for their specific environments.

It is possible that this material may contain reference to, or information about, IBM products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such IBM products, programming or services in your country.

Trademarks

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries.

BladeCenter*
DB2*
DS6000*
DS8000*
ECKD
FICON*
GDPS*
HiperSockets
HyperSwap
IBM z13*
OMEGAMON*
Performance Toolkit for VM
Power*
PowerVM
PR/SM
RACF*
Storwize*
System Storage*
System x*
System z*
System z9*
System z10*
Tivoli*
zEnterprise*
z/OS*
zSecure
z/VM*
z Systems*

* Registered trademarks of IBM Corporation

The following are trademarks or registered trademarks of other companies.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.
Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.
Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.
IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.
ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.
Java and all Java based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.
Linear Tape-Open, LTO, the LTO Logo, Ultrium, and the Ultrium logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and
Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
OpenStack is a trademark of OpenStack LLC. The OpenStack trademark policy is available on the OpenStack website.
TEALEAF is a registered trademark of Tealeaf, an IBM Company.
Windows Server and the Windows logo are trademarks of the Microsoft group of countries.
Worklight is a trademark or registered trademark of Worklight, an IBM Company.
UNIX is a registered trademark of The Open Group in the United States and other countries.

* Other product and service names might be trademarks of IBM or other companies.

Notes:

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

This information provides only general descriptions of the types and portions of workloads that are eligible for execution on Specialty Engines (e.g., zIIPs, zAAPs, and IFLs) ("SEs"). IBM authorizes customers to use IBM SE only to execute the processing of Eligible Workloads of specific Programs expressly authorized by IBM as specified in the “Authorized Use Table for IBM Machines” provided at www.ibm.com/systems/support/machine_warranties/machine_code/aut.html (“AUT”). No other workload processing is authorized for execution on an SE. IBM offers SE at a lower price than General Processors/Central Processors because customers are authorized to use SEs only to process certain types and/or amounts of workloads as specified by IBM in the AUT.

"Junior Woodchuck"?

- "The Guidebook contains information on lost treasure, a complete survival guide, extensive historical and technical information […] However, it does not contain information that a Junior Woodchuck is already supposed to know […] nor does it contain information on allegedly non-existent things"

- "Information is readily available by searching the extensive index;
  a key skill of a Junior Woodchuck is being able to retrieve information quickly from the Woodchuck book in the midst of a dangerous situation,
  such as **a bear attack**,
  an **earthquake**,
  falling out of an airplane sans parachute,
  or being **swallowed by a crocodile**. "

© 1954, Walt Disney Corporation

http://disney.wikia.com/wiki/Junior_Woodchucks_Guidebook

Agenda

- RACF for z/VM – the Short Version
- RACF Database – Error Recovery and Utilities
- RACF Database Repair – Use Cases
- Best Practices and Conclusion

A
VM Workshop
Original
Presentation

RACF for z/VM – The Short Version

What is RACF?

RACF structure

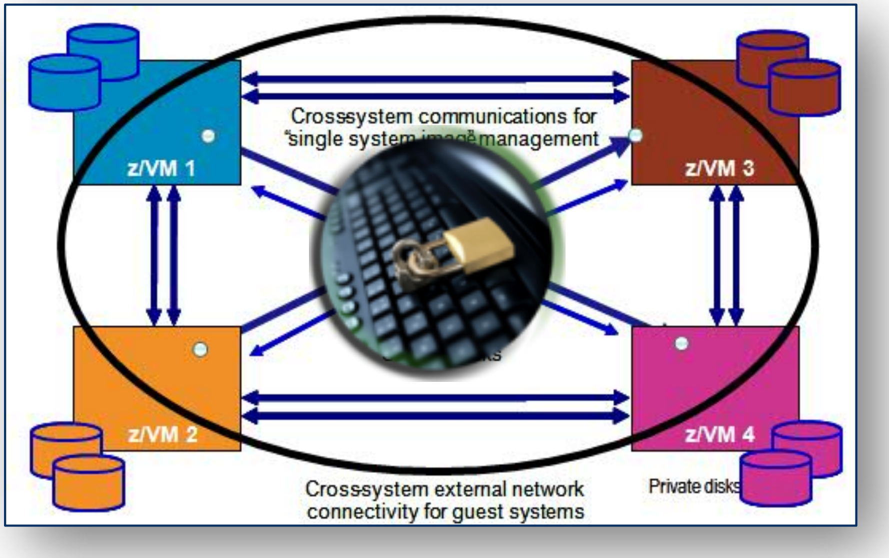
RACF database

RACF profiles

Important RACF commands for the database

Infrastructure Security with RACF for z/VM

- RACF Security Server is a priced feature of z/VM
- A **requirement** for meeting today's enterprise security requirements
- RACF enhances z/VM by providing:
  - Extensive auditing of system events
  - Strong Encryption of passwords and password phrases
  - Control of privileged system commands
  - Extensibility in z/VM environments clustered through Single System Image
  - Controls on password policies, access rights, and security management
  - Security Labeling and Zoning for multi-tenancy within a single LPAR (or across a cluster)


Crosssystem communications for "single system image" management
z/VM 1
z/VM 3
z/VM 2
z/VM 4
Private disks
Crosssystem external network connectivity for guest systems

- RACF for z/VM is an **integral component** of z/VM's *Common Criteria evaluations (OSPP-LS at EAL 4+)*

What is RACF?

(For anyone who wandered in by accident)

- **Resource Access Control Facility (RACF) is a software tool for use by:**
  - Security administrators, and
  - Auditors

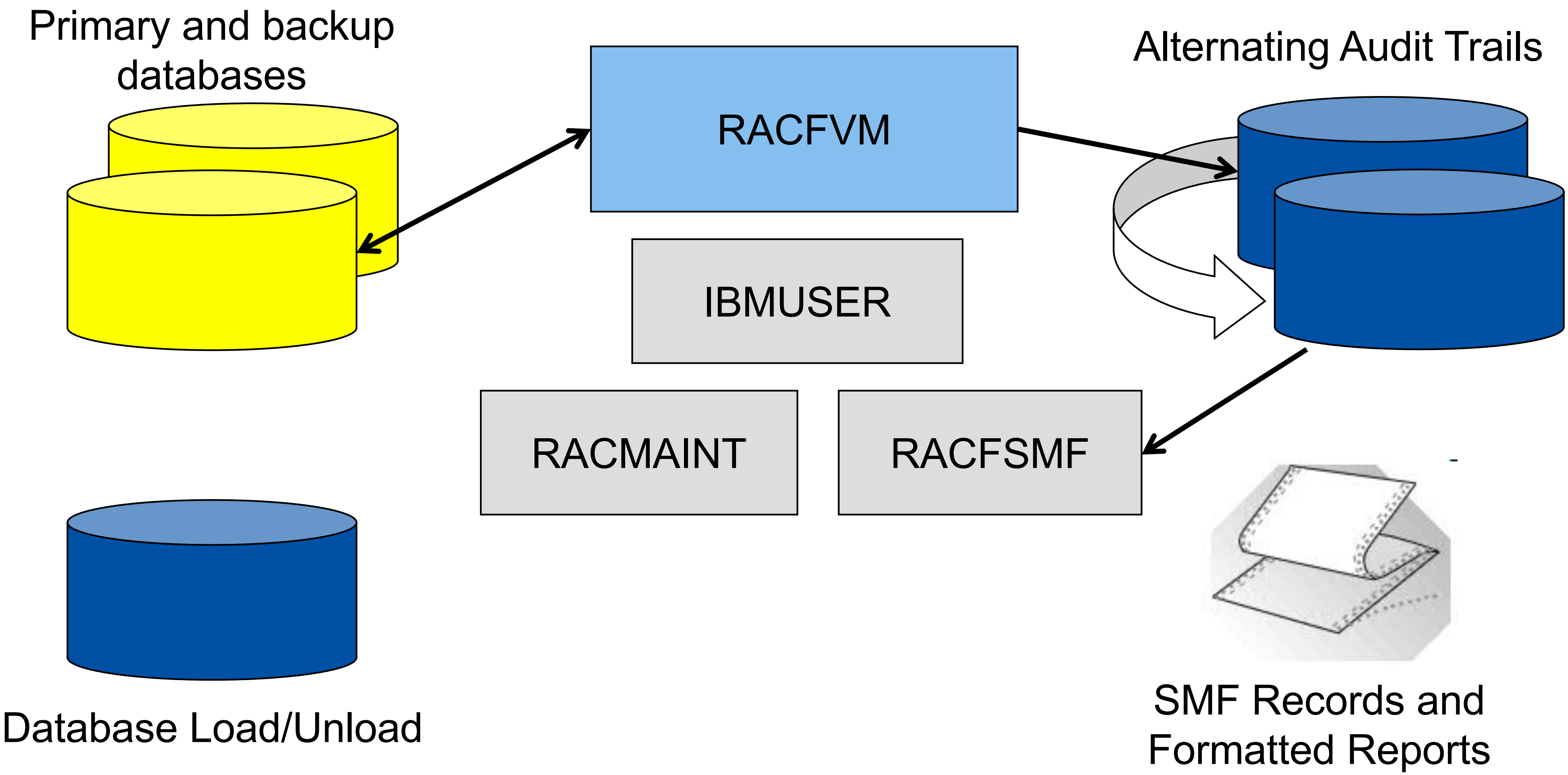
- **RACF is used to implement and monitor the implementation of the installation's security policies on z/OS and z/VM systems.**

- ***End user interaction with RACF is minimized, by design.***

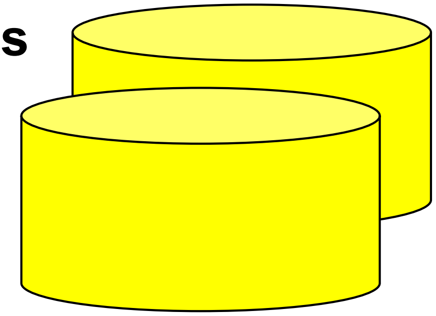
- **RACF answers the question: "*Does user abc have access to resource xyz?*"**

RACF for z/VM >> Structure

(What is this thing, exactly?)



RACF for z/VM >> The Database



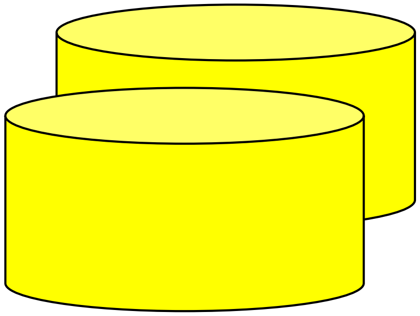
- **Where your security policy is stored – protect at all costs**
  - Contains **RACF profiles** and **system-wide options**
  - Built for speed

- **Manage with RAC commands (or panels), and utilities**
  - Control access to resources
  - Define users and groups
  - Manage system access
  - Establish accountability (audit settings)
  - Delegate authority

- **DASD can be shared across LPARS/systems**
  - Including with z/OS /* if you're brave */
  - With or without a Single System Image cluster

- **Can be accessed by z/VM LDAP (port of IBM z/OS Tivoli Directory Server)**

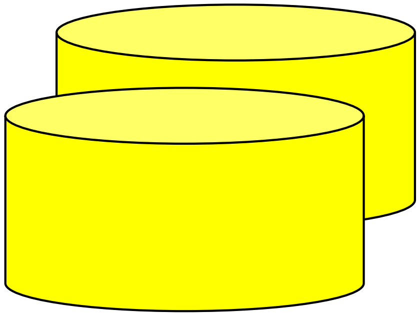
RACF for z/VM >> The Database (Priming)



- **Database is initially based on CP User Directory**
  - User directory lists every user and minidisk
- **Run EXEC RPIDIRCT against the user directory**
  - May need to make some changes beforehand to USER DIRECT, or afterwards to the SYSUT1 file
    - NOLOG passwords
    - Unacceptable characters in user IDs
    - ACIGROUP statements
    - Group names on POSIXGROUP statements that are duplicates except for case
    - OpenExtensions-related entries added by the DIRPOSIX EXEC
- The file created, **RPIDIRCT SYSUT1**, lists all the commands needed to insert the directory into the RACF database … every ADDUSER, RDEFINE VMMDISK, ADDGROUP or RDEFINE SURROGAT.

RACF and Resources >> Profiles

- **The RACF Database contains:**
  - **Profiles**, which contain the data that RACF uses to perform its identification, authentication, access control, and logging functions.
  - **Index entries**, which allow for the rapid location of profiles
  - **Control information**, such as the:
    - Byte allocation mask, which tracks the use of space within the RACF DB
    - System options, such as what classes are active
    - Templates, which define the format of the RACF data DB
    - Synchronization information to manage the operation of a shared RACF DB



RACF and Resources >> Profiles

- **There are five types of profiles:**
  - **User:** Defines the characteristics of users
  - **Group:** organize users into collections for simpler administration
  - **Connections:** Establish the relationship between users and groups
  - **Data set:** Characteristics (ACL, logging, ownership) of data sets *(if sharing with z/OS)*
  - **General resource:** Characteristics of everything other than data sets, partitioned into classes (some IBM defined, some client defined)

- **Profiles describe protection of resources**
  - Maintained by security administrator and/or users
  - Identifies owner of profile
  - Universal access authority
  - Access lists, Logging information, Security classification
  - Notification settings, “warning” indicators, and Access statistics

- Grouped into “classes” as defined in **Class Descriptor Table**
  - Can be de/activated, audited, etc. as a group using SETROPTS
  - Customers can add their own classes

RACF and Resources >> Profiles

- RACF checks for individual profiles first, then falls back to generic profiles.

- General Resource Profiles
  - Common definitions put to use in place of defining discrete entries for every class
  - Need to enable generic profiles for a given resource class
  - `SETROPTS GENERIC(class)`

- **General resource classes are defined in the class descriptor table**
  - IBM defines over 200 classes in the IBM CDT
  - Clients can define their own classes in the installation-defined class portion of the CDT

- Allows for mass definition without too much hassle

Commands to Know – If Sharing the Database

- `SETROPTS … REFRESH`: reaccess information in regards to this particular profile / class / resource / database. (This is true even if the database is not being shared.)

- In z/VM 5.4 and z/VM 6.1, you needed issue this command for **every** system sharing the database (z/VM and/or z/OS)
  - A lot of this is automated in a Single System Image cluster

- If multiple service machines exist on a single LPAR (RACFVM01 and RACFVM02), SETROPTS REFRESH must be issued for **each service machine as well**.
  - A lot of this is automated in a Single System Image cluster

- Re-IPL will obviate the need for a database refresh.

Commands to Know – Deactivating RACF

- **Disabling the database**: `RVARY … INACTIVE`
  - Important for performing maintenance on the database
  - Enables "Failsoft Processing" – system operator has to approve any requests that would normally be passed to RACF.
  - Command is not automatically propagated to other systems sharing the database
  - Users cannot logon while database is INACTIVE.

- **Deactivating RACF**: `SETRACF … INACTIVE`
  - E.g., Performing maintenance on RACF itself.
  - CP handles all authorization requests as it would prior to installing RACF.
  - Does not use failsoft processing; no RACF activity takes place.

RACF Database – Error Recovery and Utilities and Use-Cases

Error Recovery

Summary of the Utilities

Error Recovery

(This slide gleefully stolen from Bruce Hayden – thanks, Bruce!)

- RACF has built-in redundancy
  - 2 databases, a primary and a back-up
  - 2 sets of code disks
  - 2 servers (RACFVM for production; RACMAINT for test)

- Two database volumes, primary and back-up
  - Updated in parallel
  - Use the RVARY command to switch (then you may need to repair the primary)

- Please note:
  - Default location of the two databases is on the same volume!
  - May wish to have databases on separate volumes
  - SSI cluster shares the database (RDEVICE SHARED MWV) on a fullpack minidisk
    - ECKD only for sharing
    - Each database to its own separate disk as appropriate

Error Recovery

- Only the active System Operator can **XAUTOLOG** RACFVM or RACMAINT
- Only **RACFVM**, **RACMAINT**, or **OPERATOR** can log onto the system if RACF has abended
  - Using the passwords in the CP Directory
- If logged directly onto RACFVM, issue:
  - **`CP IPL 490`**
  - **`RACSTART`**

RACF for z/VM >> Utilities

- **RACFCONV** – RACF Database Conversion
  - Used to update the template in use by the RACF database
  - Usually run when applying a New Function PTF or installing a new z/VM level
  - Invokes a program called IRRMIN00 under the covers
    - MIN00.OUTPUT

- **RACUT200** – Database Verification Utility (IRRUT200 to its friends)
  - Usually used to make copies of existing databases
  - Also validates current structure of an existing database
  - Copy and verification only; nothing more
  - Always run this before RACFCONV.

- **RACUT400** – Database Extend Utility (IRRUT400 to its friends)
  - Most common use: copy database to larger or smaller target volume
  - Also reorganizes and restructures the database itself
  - So occasionally used for cleaning up profile errors

- **BLKUPD** – Block Update Utility. We'll get to this later.

RACF Database Repair – Use-Cases

Make a Copy of the Database

Validate Database Integrity

Upgrade the Database (2)

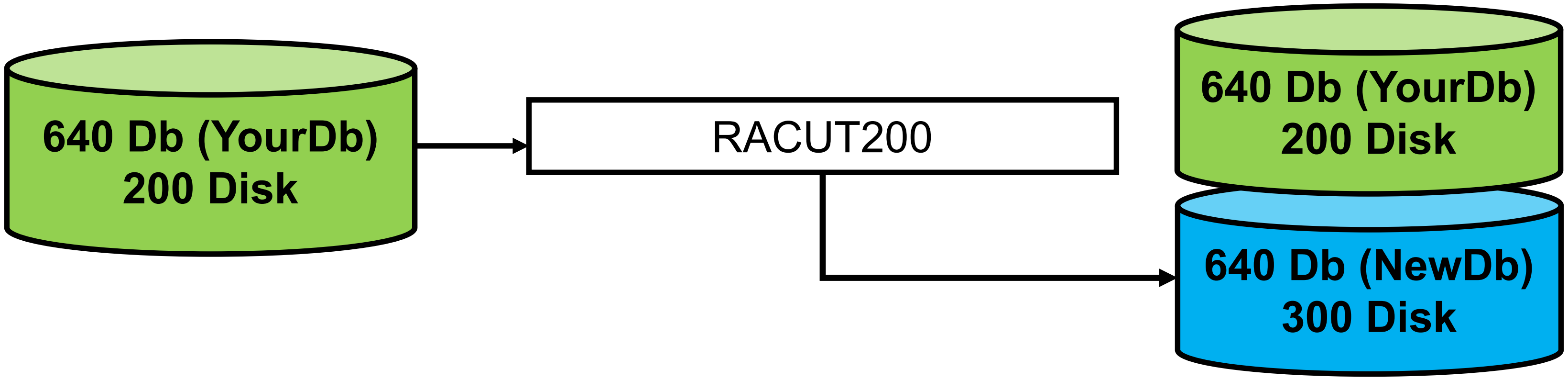
Repair a RACF Database

If Things Go Really Wrong …

Use-Case: Make a Copy of the Database

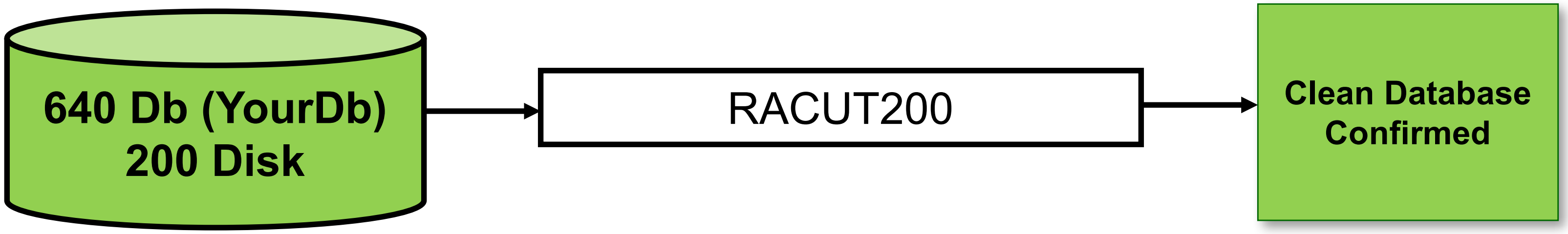
- Your Utility is: RACUT200
  - Enter **RACUT200.**
  - Reply **NO** to the 'Do you want to Verify a RACF database?' prompt.
  - Reply **200** to the 'Enter the Input device address' prompt.
  - Reply **300** to the 'Enter the Output device address' prompt.
  - Reply **YES** to the 'Do you wish to continue?' prompt.
  - Messages RPIRND003E and IRR62009I can be ignored.
  - Return code from 'IRRUT200' = 0 should be issued if successful.

- *Note: while reserve/release should quiesce database during copy, copying a live database may lead to unpredictable results – things might yet change under the covers!*



Use-Case: Validate Database Integrity

- Your Utility is: RACUT200 (again)
  - Enter **RACUT200**.
  - Reply **YES** to the 'Do you want to Verify a RACF database?' prompt.
    - If a RACVERFY FILE input file exists, you will be given the option to reuse it or overlay it. If a RACVERFY FILE does not exist, one will be created and XEDIT will be entered. Type **FILE** when editing is complete.
  - Reply **200** to the 'Enter the Input device address' prompt.
  - Press **Enter** to bypass copy.
  - Reply **YES** to the 'Do you wish to continue?' prompt.
  - Messages RPIOPN003E and IRR62003I can be ignored. (You may also get messages DMSLOS013E and IRR62064I.)
  - Return code from 'IRRUT200' = 0 should be issued if successful.



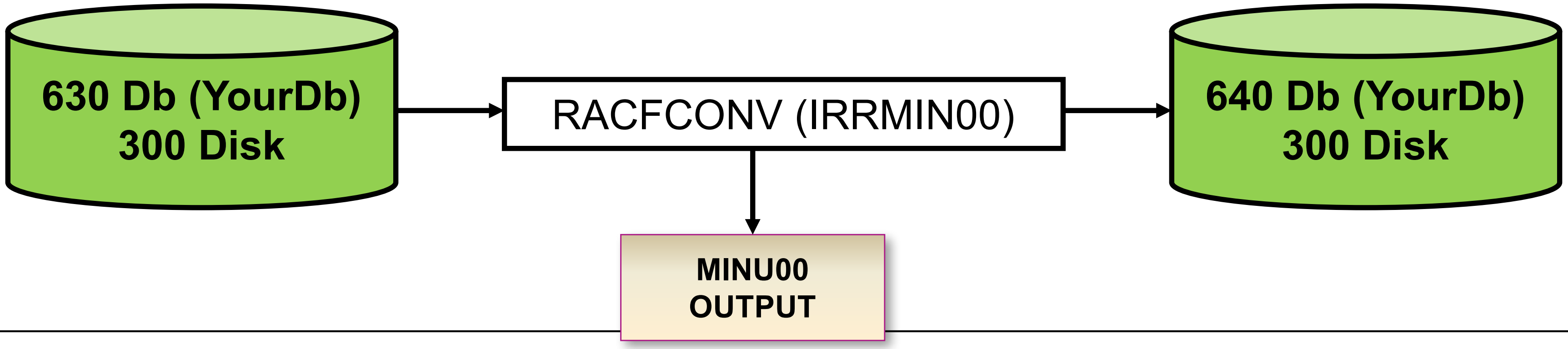
  - The IRRUT200 output report will be sent to your virtual printer.

Use-Case: Upgrade a Standalone Database

- Your utility is: RACFCONV
  - **Shut down** the RACFVM virtual machine. (**FORCE** from **OPERATOR**)
  - **LOGON RACMAINT** (you'll want to upgrade from here)
  - Enter **IPL 190**, enter **RACFCONV, then** Press **Enter**
  - Select your volume, e.g. **300,** then Enter **yes**
- **Notes:**
  - Validate Database integrity before converting (RACUT200)
  - **RC=4** means the database template did not need to be updated. You may proceed

```
RACFCONV
An Error occurred during 'IRRMIN00' processing
Return code from 'IRRMIN00' = 4
Ready(00004);
```

  - **Update both primary and backup databases** before IPL'ing RACF



630 Db (YourDb)
300 Disk

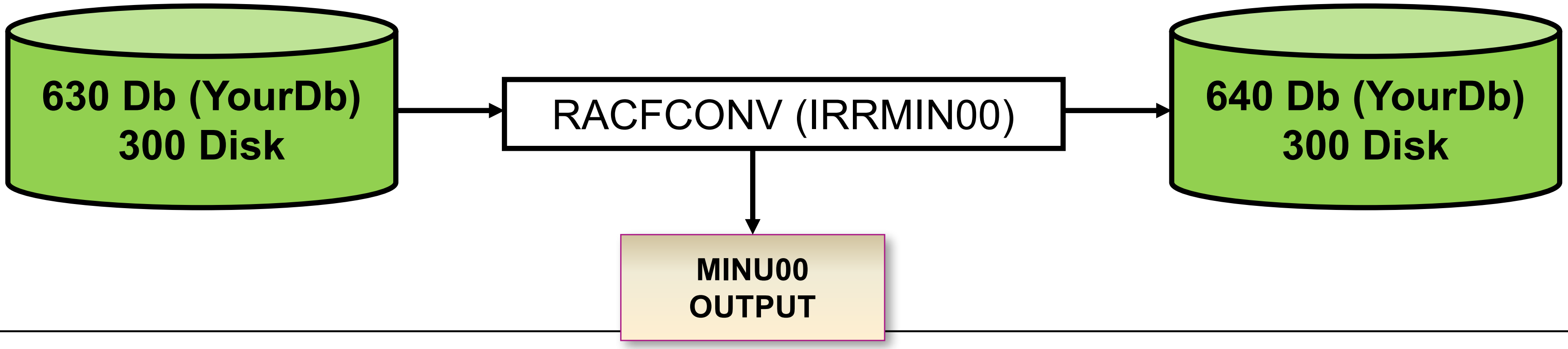
RACFCONV (IRRMIN00)

640 Db (YourDb)
300 Disk

MINU00
OUTPUT

Use-Case: Upgrade a Shared RACF Database

- Validate the links for your shared database to RACMAINT, whether in SSI or not
- Applying service:
  - SSI: from MAINT6n0, from only one member to start
  - Non-SSI: apply service only to one system to start
- Shut down all RACFVM virtual machines.
- Follow steps on previous slide (Upgrade first database – RACFCONV)
- **XAUTOLOG** RACMAINT on all other systems associated with this database
- **PUT2PROD** as appropriate for remaining systems
- **FORCE** RACMAINT and **XAUTOLOG** RACFVM

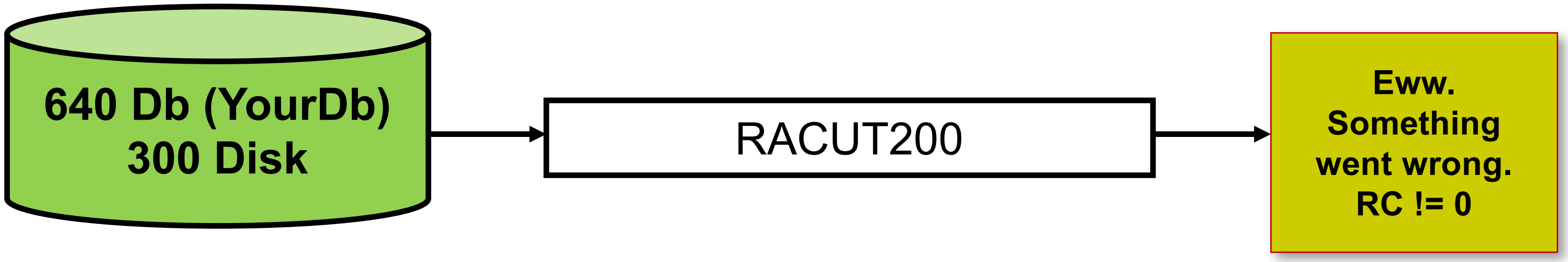


Use-Case: Repair a RACF Database

- But maybe RACUT200 reports some errors
  - Can we fix it? *Yes we can!*
  - *… maybe!*

- Your actions may vary based upon the sort of errors you're seeing
  - *RACFVM Diagnosis Guide, Chapter 5 – Troubleshooting Your Database*
  - This presentation will be a starting point, but follow directions and instructions to the letter (No, don't skip steps.) (Yes, read everything twice.)

- First item of business: is this a production database?
  - **If yes**: back it up, attach it to a test z/VM system, do the repairs from there.
  - **If no:** back it up also, but proceed against target system.

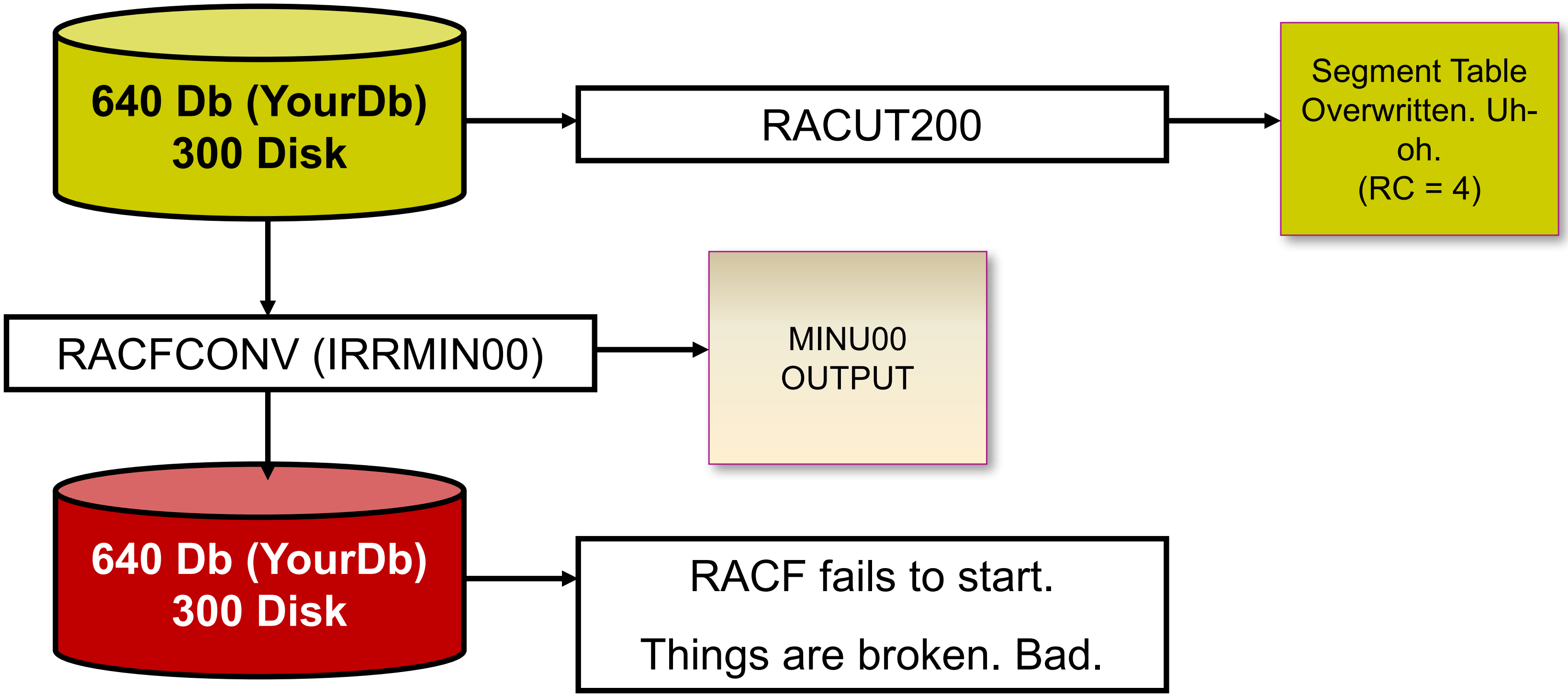


Use-Case: Repairing a RACF Database

- You "may" want to force off other machines sharing this database for this next bit.

- Note that some errors (duplicate profile definitions) aren't catastrophic
  - – Database keeps running, just looks weird
  - – But like a melting lake of ice, it'll get complicated later
  - – **Best to fix sooner rather than later**

How to corrupt your database:



640 Db (YourDb)
300 Disk

RACUT200

Segment Table Overwritten. Uh-oh.
(RC = 4)

RACFCONV (IRRMIN00)

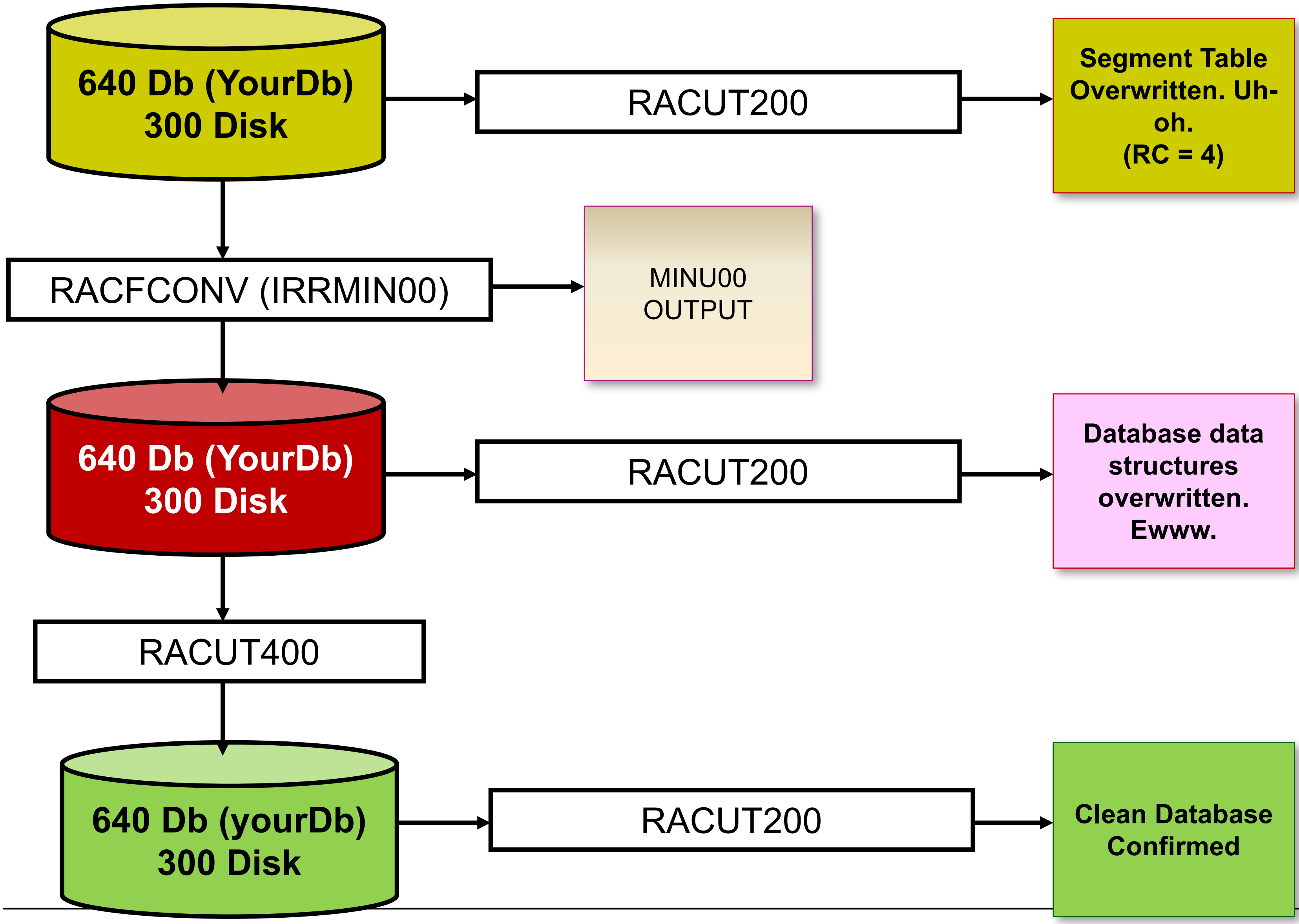
MINU00 OUTPUT

640 Db (YourDb)
300 Disk

RACF fails to start.

Things are broken. Bad.

How to fix it: measure twice, cut once

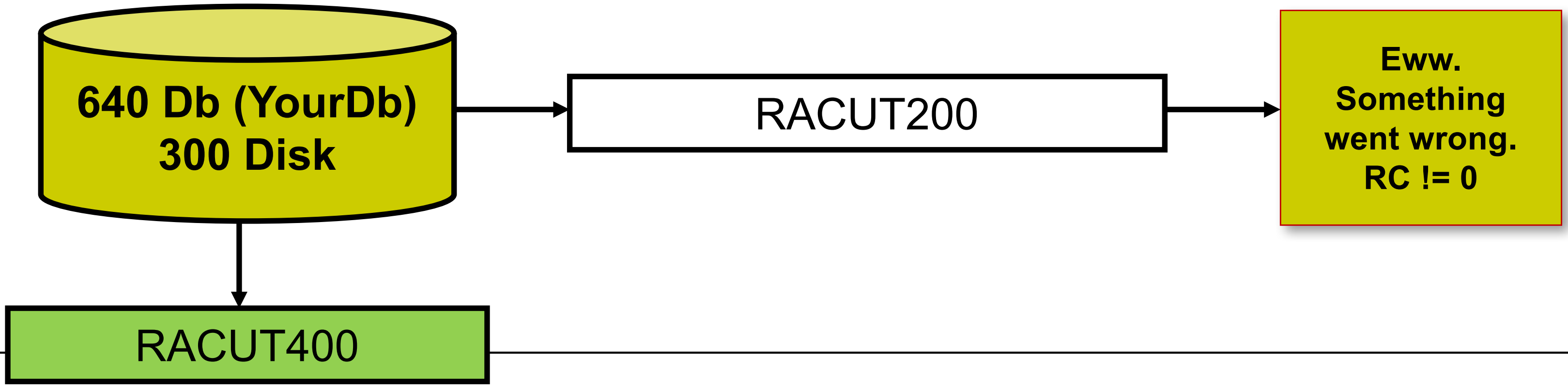


Use-Case: Repairing a RACF Database

- You may want to force off other machines sharing this database for this next bit.

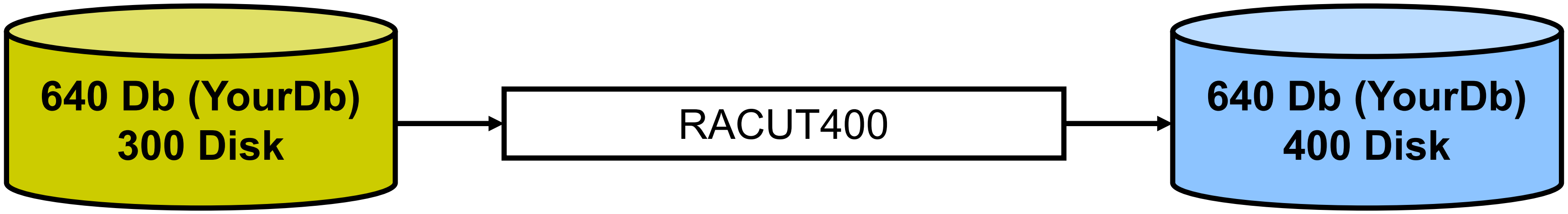
- Note that some errors (duplicate profile definitions) aren't catastrophic
  - Database keeps running, just looks weird
  - But like a melting lake of ice, it'll get complicated later
  - Best to fix sooner rather than later

- Your utility: **RACUT400**



Use-Case: Repairing a RACF Database

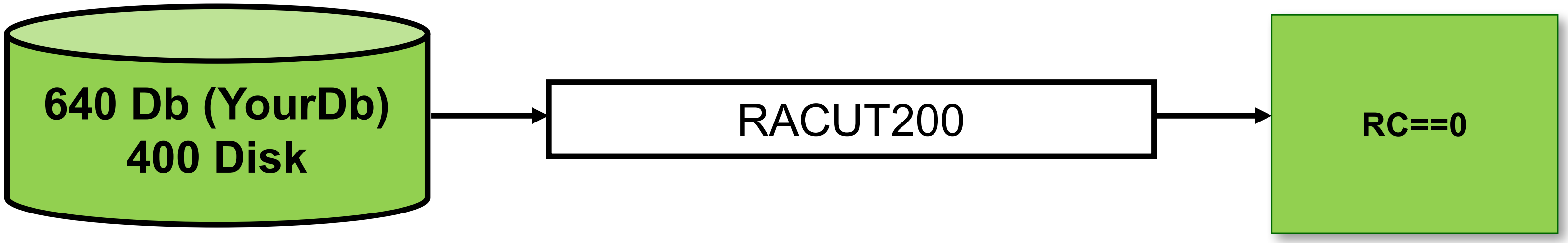
- Enter **RACUT400**
  - Reply **COPY** to the 'Enter SPLIT or MERGE or COPY or QUIT' prompt.
  - Reply **200 or 300** to the 'Enter the single input device address' prompt.
  - Reply **400** to the 'Enter the single output device address' prompt.
  - Reply **YES** to the 'Do you wish to continue?' prompt.
  - Reply **CONT** to enter parameters.
  - If executing against an offline or unshared database, Reply NOLOCKINPUT to the first ‘Enter Next Parameter’ prompt. Otherwise, reply **LOCKINPUT** to the first 'Enter Next Parameter' prompt.
  - Reply **END** to the second 'Enter Next Parameter' prompt.
  - Return code from 'IRRUT400' = 0 should be issued if successful



Use-Case: Repairing a RACF Database

- **Run RACUT200 against the database again.**
  – It's the only way to be sure.

- If it was a structural problem, then RACUT400 reorganized the database and gave a clean return code. Hooray!

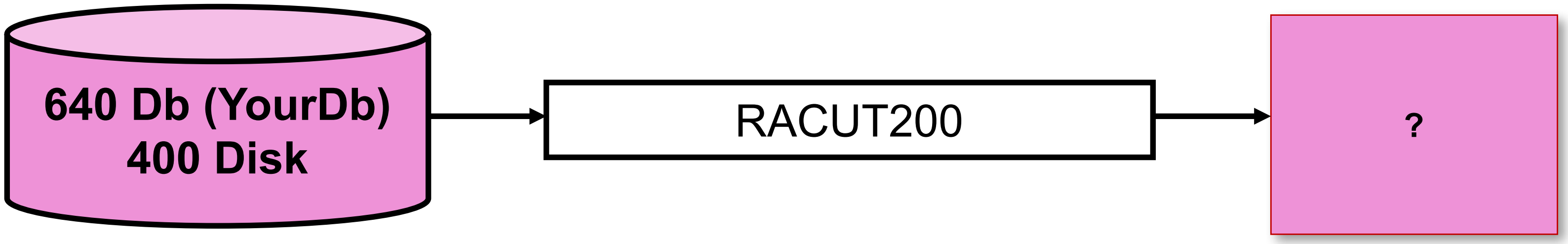


640 Db (YourDb)
400 Disk

RACUT200

RC==0

- If not ….



640 Db (YourDb)
400 Disk

RACUT200

?

Use-Case: If Things Go Really Wrong …

- If the problem is profiles,
  - You may want to delete and restore the profile in question
    - **RAC DELUSER, DELGROUP, or RDELETE** /* to remove a profile */
    - **RAC ADDUSER, ADDGROUP, or RDEFINE** /* to add it back */
  - Easier than messing about with internals
  - Standard RAC commands work (RDELETE, etc.)

- If the problem is huge, or systemic, you have one of three options.
  1. **Restore** from back-up
  2. **Rebuild the database**
  3. **BLKUPD** (more on this in a moment)

- Not every database can be saved
  - Sad, but true
  - If you let corruption continue, eventually things will break hard
  - Remember to validate integrity before updating database with new template
    - Template changes may reorganize database structure!

Use-Case: Rebuilding the Database

- **RACF Database Unload Utility**
  - **RACFDBU** / IRRDBU00
  - The Database Unload Utility reads every profile and creates records of various types.
  - This output can be viewed directly or used as input to another program.
  - Also creates SQL-mappings of the database output
  - Can be used to view database contents in a more friendly way …
  - … make changes …
  - … and reload into a test database for repair purposes

- **RPIDIRCT**
  - Good way to "start over from scratch"
  - Processes USER DIRECT into RACF definitions and permissions
    - Can also execute against MYUSER DIRECT files, in case just one VM is missing
    - Guidance for the "re-add" portion of work from previous slide
  - Handles a lot of the "system definition" work
    - Does **not** cover password policy and RACF configuration options
  - Also useful to compare against Database Unload output – what changed?

Use-Case: Bit-Level Surgery with BLKUPD

- BLKUPD (Block Update)
  - The **BLKUPD** utility can be used to make manual adjustments to the RACF database.
  - Note that **BLKUPD** is not recommend for use with BAM errors (use **RACUT400**), should not be used in periods of high activity, and should only be used after using **RACUT200** to determine what problems are left.

- ***Note***: *this utility is akin to performing brain surgery on the RACF database*

Before you use the BLKUPD command, you should be very familiar with the RACF database and its configuration because improper usage of BLKUPD can result in damage to the RACF database. (See “Format of the RACF Database” on page 53 of the *z/VM RACF Diagnosis Guide*.) In general, use this command **only when directed to by the IBM support center.**

You should **read and understand the pages on the format of the database before entering the BLKUPD command.** Then, before you begin to use the BLKUPD command to perform updates to your RACF database, **we recommend your trying to use one of the RACF commands to alter or delete the entry in question.**

BLKUPD >> Steps

- Decide which database you want to work with, and enter BLKUPD
- Decide which block (logical data location) on the database you want to work with.
  - If needed, use the LOCATE subcommand to assist you in finding the specific block.
  - ***If you've reached this point, IRRUT200 output should provide guidance in this.***
- Enter the **READ** subcommand, specifying either UPDATE or NOUPDATE
- Enter the subcommands of **READ** necessary to accomplish your task
- Issue the **END** command to end the utility.

BLKUPD >> Sample

```
racf
RPITMP001I RACF/VM SESSION ESTABLISHED. TO TERMINATE ENTER "END"
RPITMP002I ENTER RACF COMMAND OR "END" TO EXIT
blkupd
BLKUPD:
read x'F1000' update
BLKUPD:
display entry(BWH191) class(VMMDISK)
OFFSET COMP. ENTRY NAME RBA COUNT
00E 000 VMMDISK -EKH551 0000000F8200
7F4 009 VMMDISK -CHUCKE 0000000F8300
next
80B 009 VMMDISK -BWH191 0000000F8400
delete
80B 009 VMMDISK -MNT190 0000000F8500
end
IRR63027I ENTER SAVE OR NOSAVE
save
IRR63009I DISPLAY ended. Changes saved.
BLKUPD:
end
IRR63027I ENTER SAVE OR NOSAVE
save
IRR63013I READ ended. Block saved.
BLKUPD:
end
RPITMP002I ENTER RACF COMMAND OR "END" TO EXIT
end
RPICMD003I RACF/VM COMMAND SESSION COMPLETE
```

Use-Case: Communication Breakdown

- **Very infrequently, RACF may RESERVE the database and not let go**
  - There are a lot of moving parts under the covers of RACFVM operations
  - Simulation of MVS code inside of CMS
  - RACF doesn't like to take chances with someone stealing the database
    - It's your security policy, after all!

- **Recommendation**:
  - Don't take a dump right away (you're looking for something that didn't happen)
  - Instead, do a QUERY DASD DETAILS from each LPAR sharing the database
    - QUERY DASD RESERVE as well
  - And check your OPERATOR console for I/O errors

- **Consider**:
  - Are the channel paths still there from your system to the DASD?
  - Have there been I/O errors recently, even if they seemed incongruous at the time?
  - Was someone doing a concurrent upgrade on your DS8800?

- Engage IBM Service as appropriate

Best Practices and Conclusion

RACF Database >> Best Practices (1/2)

- **Make a policy for creating and validating copies** of RACF databases
  - You need both a valid primary and valid back-up to start an SSI cluster
  - If you've broken everything, you're in trouble

- **Keep a safe, solid, and current reserve copy** of your RACF database
  - Distinct from primary and backup volumes!
  - Often easier to swap in a new volume than repair an existing one

- **Validate and integrity-check databases before an upgrade** of any sort
  - If you're asked to issue RACFCONV, always issue RACUT200 first!

RACF Database >> Best Practices (2/2)

- **Segment judiciously**
  - Modern performance means there's little need to shard the RACF database (or the RACFVM virtual machines)

- **Share cautiously**
  - Database can be shared between z/VM systems (SSI or non-SSI) on ECKD DASD
  - RACFVM databases can be shared with z/OS, too
  - … if templates match, and if security contexts are in line with one another
    - Not every system has the same security needs!

- **Automate extensively** around RACF start-up problems
  - Operations Manager to swap in known valid backup copies of the database
  - Allows for start-up; processing and repair work can continue
  - Products like zSecure for RACFVM can help you manage RACF once you're up and running

Conclusion

- **RACF provides enterprise-level security for your z/VM system**
  - Requirement to meet System Integrity Statement
  - Requirement to meet Common Criteria evaluation
  - Encrypts your passwords
  - Audits your system
  - Provides multi-tenancy

- **Don't panic if something goes wrong with your RACF system**
  - There are tools at your fingertips
  - Advice online, on the RACF mailing lists, at RACF User Groups

- **Do have a plan**
  - Make copies at regular intervals
  - Validate your databases
  - Automate where possible (especially around system recovery)

For More Information …

- http://www.vm.ibm.com/security/
- https://www-03.ibm.com/systems/z/os/zos/features/racf/vm.html
- https://www-03.ibm.com/systems/z/os/zos/features/racf/resources.html

- **With special thanks to**:
  - Ian Broadbent
  - Robert Hart
  - Bruce Hayden
  - Mary Hottenstein
  - Mary Stefos

Contact Information:

Brian W. Hugenbruch, CISSP
IBM z Systems Virtualization Security
bwhugen at us dot ibm dot com
@Bwhugen

RECOGNIZED SPEAKER/PRESENTER
IBM
Imparting knowledge
IBM PROFESSIONS

Dank u
Dutch

Merci
French

Спасибо
Russian

Gracias
Spanish

شكراً
Arabic

धन्यवाद
Hindi

감사합니다
Korean

Tack så mycket
Swedish

תודה רבה
Hebrew

Obrigado
Brazilian
Portuguese

谢谢
Chinese

Dankon
Esperanto

Thank You

ありがとうございます
Japanese

Trugarez
Breton

Danke
German

Tak
Danish

Grazie
Italian

நன்றி
Tamil

děkuji
Czech

ขอบคุณ
Thai

go raibh maith agat
Gaelic

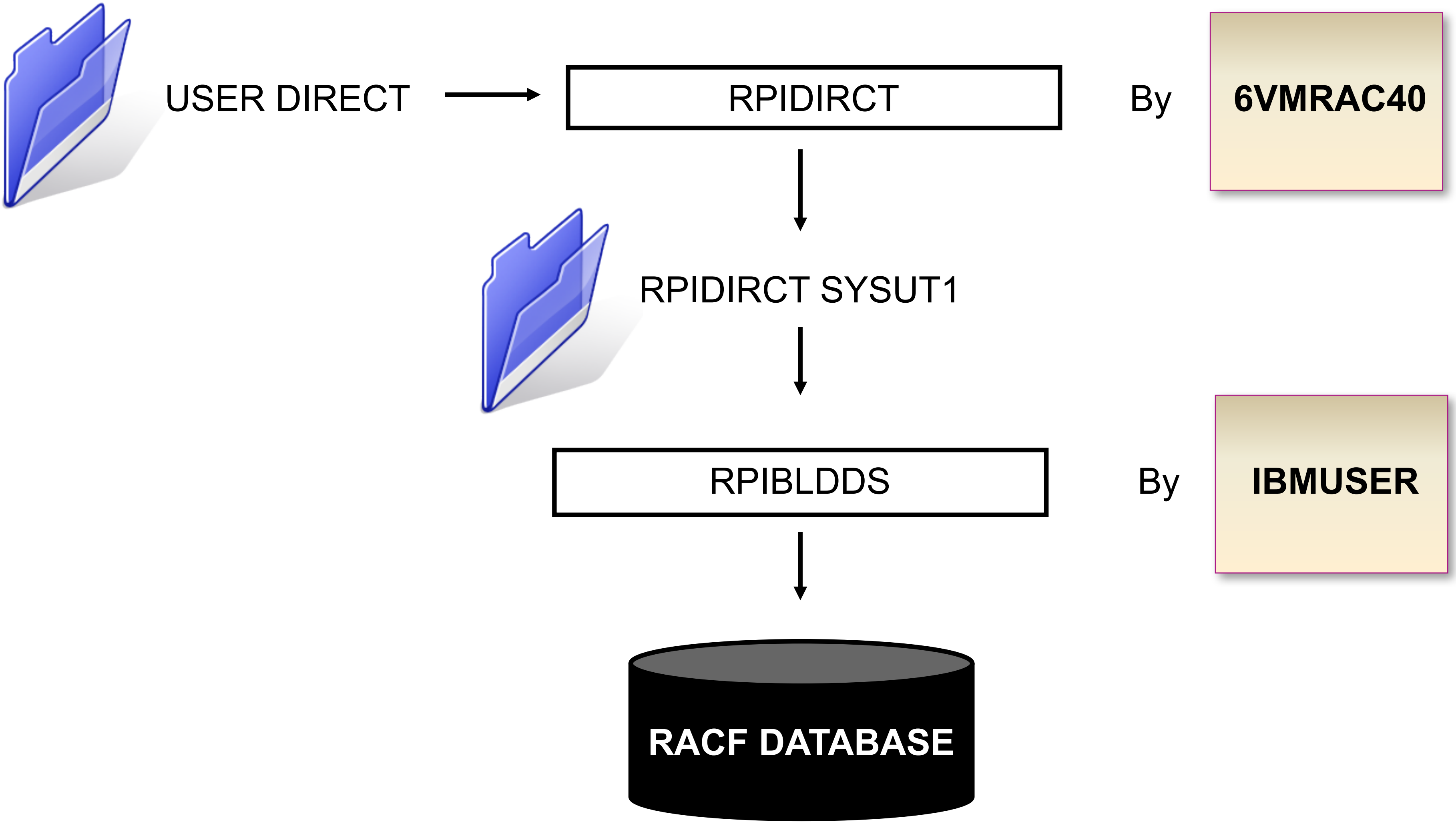
#vmworkshop #zVM #IBMz

Bonus Content:
Building a new RACFVM Database

Quick and dirty version:

Refer to the RACF Program Directory for the real goods!

Where a RACF Database comes from …



USER DIRECT → RPIDIRCT By 6VMRAC40

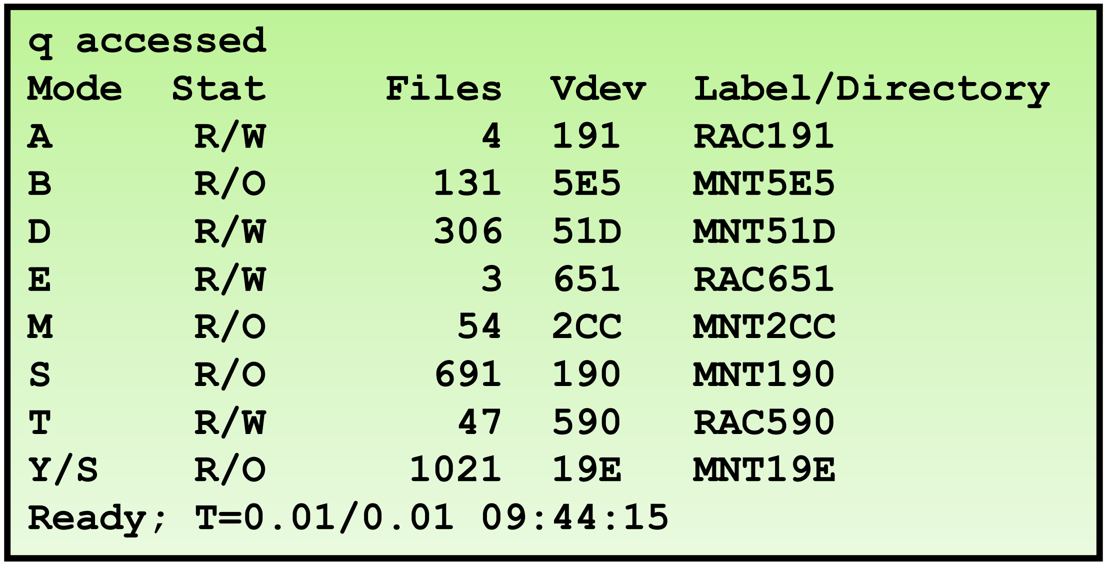
RPIDIRCT SYSUT1

RPIBLDDS By IBMUSER

RACF DATABASE

Step 01: Preprocess the CP directory

- On your second-level system, **LOGON 6VMRAC40**
- `ACCESS 651 E`
- `LINK MAINT 2CC 2CC RR READ`
- `ACCESS 2CC M`
- `QUERY ACCESSED` should display the following:



```
q accessed
Mode   Stat      Files  Vdev   Label/Directory
A        R/W         4  191    RAC191
B        R/O       131  5E5    MNT5E5
D        R/W       306  51D    MNT51D
E        R/W         3  651    RAC651
M        R/O        54  2CC    MNT2CC
S        R/O       691  190    MNT190
T        R/W        47  590    RAC590
Y/S      R/O      1021  19E    MNT19E
Ready; T=0.01/0.01 09:44:15
```

Step 01: Preprocess the CP directory

On 6VMRAC40:

- `RPIDIRCT USER DIRECT M  A`
  - Keep SYS1 as default group ID for now.
  - The screen is going to clear a lot.

```
/* this converts the User Directory into a series of */
/* RACFVM rules - in other words, a security policy. */
```

Step 02: Edit RPIDIRCT output

On 6VMRAC40:

- `XEDIT RPIDIRCT SYSUT1`
  - `CHANGE /PASSWORD(NOLOG)/NOPASSWORD/*`
  - `ALL /VMBATCH/`
  - `DELETE *`
  - `FILE`

- You won't want everything RPIDIRCT gives you – pick and choose based on your company's security policy
  - Most installations keep VMMDISK, VMDEV, and VMLAN
  - The above removes the VMBATCH definitions, not as common

Step 03: Load the database

XAUTOLOG RACMAINT (for your testing), and then:

- **LOGON IBMUSER**
  - Default Password of SYS1 … but you'll note, on LOGON, that it's expired!
  - Change to 999999 (or something you'll remember easily)

- Once connected to IBMUSER, link some minidisks so you can issue RACF commands … and then convert that RPIDIRCT file into an actual database:

```
LINK 6VMRAC40 651 305 RR
ACCESS 305 C
LINK 6VMRAC40 191 192 RR
ACCESS 192 B
LINK 6VMRAC40 29E 29E RR
ACCESS 29E D
RPIBLDDS RPIDIRCT
```

Step 04: Profit

You'll start seeing RPI* messages on your OPERATOR console right away

Test out your database with RACMAINT for now

- *Cut over to RACFVM when you're ready*
- *Follow the RACFVM Program Directory and other IBM guidance.*

NOLOG the IBMUSER virtual machine

- *It's a powerful and authorized RACF user, no sense leaving that door open!*

And add RACFVM to **AUTOLOG1**

- *Recommend moving everything else from AUTOLOG1 to AUTOLOG2*
- *AUTOLOG1 can xautolog AUTOLOG2 once RACFVM is running*