# SIMPLE STEPS TO CLEAN UP THE **RACF DATABASE**

Stu Henderson
5702 Newington Road
Bethesda, MD 20816

stu@stuhenderson.com
www.stuhenderson.com
(301) 229-7187

# AGENDA



I. Introduction

II. Basic Maintenance

III. Eliminating Deadwood

IV. Consolidation

V. Building A New Infrastructure

VI. Summary and Call to Action

# ABSTRACT



After years of work, and a parade of administrators, many RACF Databases can benefit from a good clean-up.

While RACF Administrators are often assigned to do this clean-up, it is one of those tasks that you learn best by doing, after which you don't need the knowledge for a long time (you hope).

In this session, Stu will show you basic, simple clean-up techniques that have proven to be both practical and beneficial.

# I Introduction



1) What Do We Mean By Clean-Up?

2) Why Do We Want To Do This?

# I Introduction



## 1) What Do We Mean By Clean-Up?

- Knowing the basic maintenance is completed
- Eliminating deadwood
- Consolidating where practical
- Having a simple structure that is easily explained to anyone, at any time

# I Introduction



## 2) Why Do We Want To Do This?

- Save space and reduce I/O
- Know only authorized profiles exist
- Be able to justify everything in DB
- Simplify administration

# II Basic Maintenance



1) Proper Backups and Restores

2) IRA

(Independent Reorganization of Alias)

3) IRRUT200 Analysis

# II Basic Maintenance



## 1) Proper Backups and Restores

- Why full pack dump/restore invites corruption
- Use IRRUT200 every night
  Do DBU at same time
- But what about recovery & restore?
- Don't forget to test RVARY procs

# II Basic Maintenance



## 2) IRA (Independent Reorganization of Alias)

- A re-org to the RACF DB

(or AIM, Application Identity Management)

- Ask your SysProg; Use RACF SysProg Manual; Ask early, schedule at her convenience
- Adds functionality, improves performance
- Moves data into index records

# II Basic Maintenance



## 3) IRRUT200 Analysis

- 200 is not just for Backups
- Additional control cards show
  - percent full
  - amount of wasted space in index blocks
  - when to re-org with IRRUT400
  - allocation map verification
- Perform analysis every 3 or 6 months

# III Eliminating Deadwood



1) Old Users, Groups, Datasets, Resources

2) Old Permit Lists, (Including Obsolete Permit List Entries), and Rules of Thumb

3) One Resource Class at a Time, One Application at a Time

4) Knowing Why Everything is There

# III Eliminating Deadwood



## 1) Old Users, Groups, Datasets, Resources

- How do you know when they are obsolete?
  - not used (last used date)
  - no members, no HLQs, no ownership
  - no permit list
  - no approval or annual re-certification

- But exceptions to all the rote checks
  - If you don't know, what's the risk of deleting?

# III Eliminating Deadwood



## 2) Old Permit Lists

- Users and groups that don't exist any more

- The "problem" with Mary

*"I've got 10 new employees starting work tomorrow who'll be helping Mary out. I need you to give them userids that can do whatever Mary can do! And right now!"*

Note: Mary currently works in Payroll, but previously worked in both HR and the warehouse.

# III Eliminating Deadwood



## 2) Permit List Rules of Thumb

- If a production application has more than 4 dataset rules, ask why. (It seldom needs to be so complicated.)
- If a production dataset or resource rule has more than 6 entries in the permit list, or more than 2 userids, ask why.
- Consider all the reasons for permitting to groups instead of userids, and for having role-based groups for security administration

# III Eliminating Deadwood



## 2) Permit List Rules of Thumb

| PROD.PAYROLL.YEAR2008.JULY.CHEX |
| --- |
| UACC(READ)<br>DATA('PAYROLL FOR JULY 2008-CHECKS') |
| GROUPA / UPDATE |
| USER25 / NONE |
| USER27 / READ |
| MARY / CONTROL |
| PAYGROUP / ALTER |
| PAYUSER / UPDATE |
| MARY1 / READ |
| MARY3 / ALTER |
| SYSPRG1 / UPDATE |
| LIKEMARY / CONTROL |

# III Eliminating Deadwood



## 3) One Resource Class at a Time

- Who is the owner?
- Is it active?
- What rules exist?
- Generics? Global entries?
- Auditing active?
- What approvals?

# III Eliminating Deadwood



## 3) One Resource Class at a Time

- The story of VTAMAPPL

# III Eliminating Deadwood



## 3) One Resource Class at a Time

- For Digital Certificate Classes, see RACF User News, Issue 72:

www.stuhenderson.com/RUGNEW72.pdf

# III Eliminating Deadwood



## 4) Know Why Everything is There

- For every User, Group, Dataset Profile, and Resource Profile and for every Option set, you want to have someone else who is responsible for:
  - approving it,
  - annually re-certifying it, and
  - having it removed at the proper time

# III Eliminating Deadwood



## 4) Know Why Everything is There

- Then you never need to wonder what some profile or option is for
- It's easier to deal with auditors
- And you get less grief, since people know that you don't approve, you only carry out what others have approved

# IV Consolidation



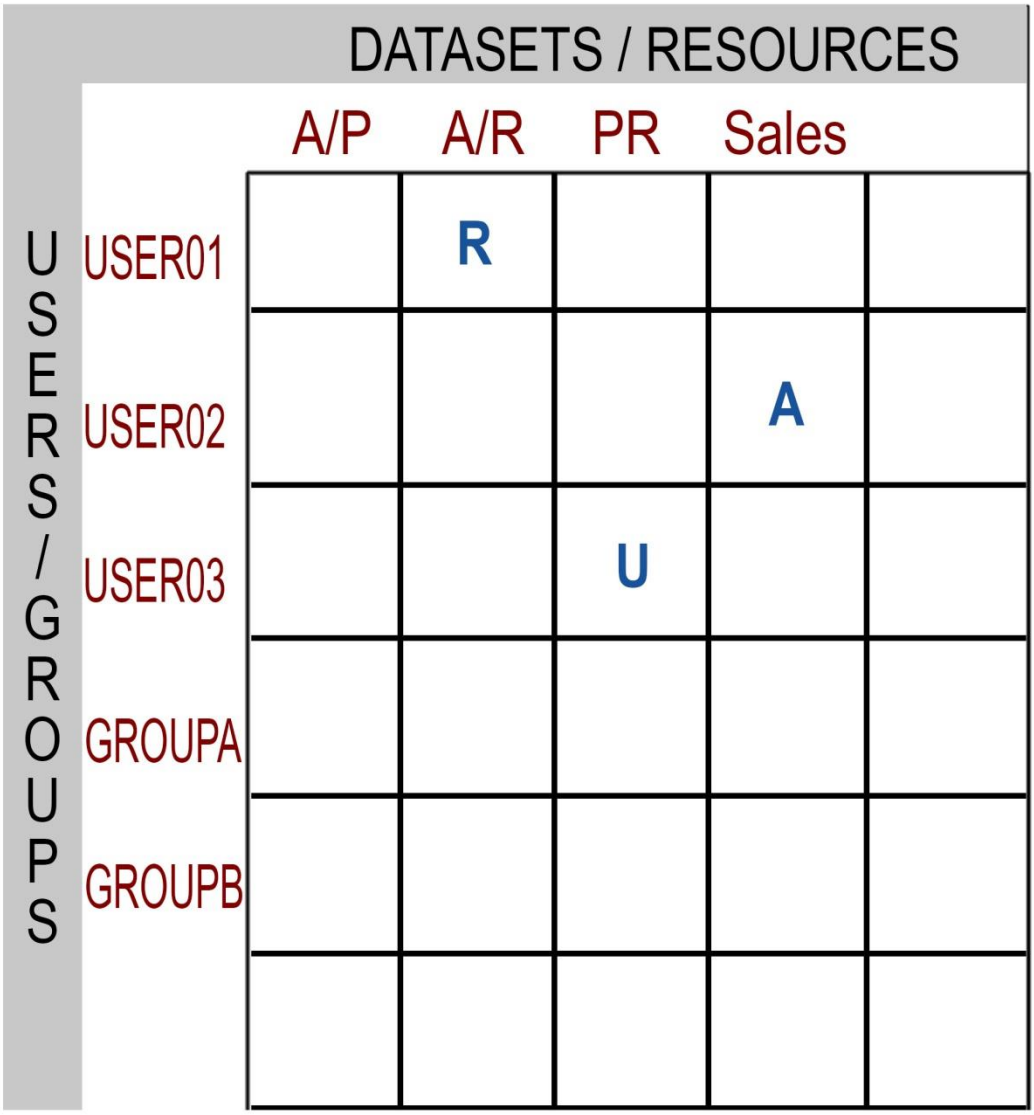
1) The Access Matrix

2) Ways To Consolidate

# IV Consolidation



## 1) The Access Matrix



# IV Consolidation



## 2) Ways To Consolidate

List three ways to take these three CICS transactions and treat them the same:

INQ1, INQ2, and INQ3

- Two Wildcards ( * and % )
- One Substitution (&)
- Grouping Classes

# V Building A New Infrastructure



Sometimes, it's not practical to clean things up, (unless you're able to run a river through the stable), so:

- You build a new infrastructure that is simpler and easier to understand and explain (and one that takes nothing away from anyone)
- All the while gently preparing your RACF so other people can peel the old stuff away

# V Building A New Infrastructure



So for each application build 2 Groups:

- One Group that allows Users to READ its data
- One Group that allows Users to: do ANYTHING to its data

# V Building A New Infrastructure



- Permit each Group appropriately to its Dataset and Resource Rules

  (This does not prevent any user from doing anything)

- Get the application's owner to approve who should be in each Group

# V Building A New Infrastructure



- Connect each User to the approved Groups

  (This still does not prevents a user from doing anything)

- Then make it someone else's job to tell you when to peel away the old mess, one application at a time

# V Building A New Infrastructure



This allows you to gradually build a database where every entry has an Approver and an Approval

Successful implementation requires:

- Only *data owners can approve* access to the data
- RACF administrators can only *implement data access* after proper approval by the data owner

Anything else is not good security,

and not auditable

# VI Summary and Call To Action



There are several threads to the clean-up process:

- Basic Maintenance
- Eliminating Deadwood and Simplifying by Consolidation
- Moving to a Proper Infrastructure with Clear Organizational Responsibility

# VI Summary and Call To Action



The first two are needed

But only the third one will last

And only the third one will give you effective, demonstrable, auditable, and understandable security

If not you, then who?

# For Further Information



1) RACF User News (back issues & subscribe)
http://www.stuhenderson.com/Newsletters-Archive.html

2) Articles on Mainframe Security and Audit
http://www.stuhenderson.com/Articles-Archive.html

3) IBM RACF Manuals

Thanks for Your Kind Attention

Questions to Stu Henderson

(301) 229-7187
stu@stuhenderson.com
http://www.stuhenderson.com/