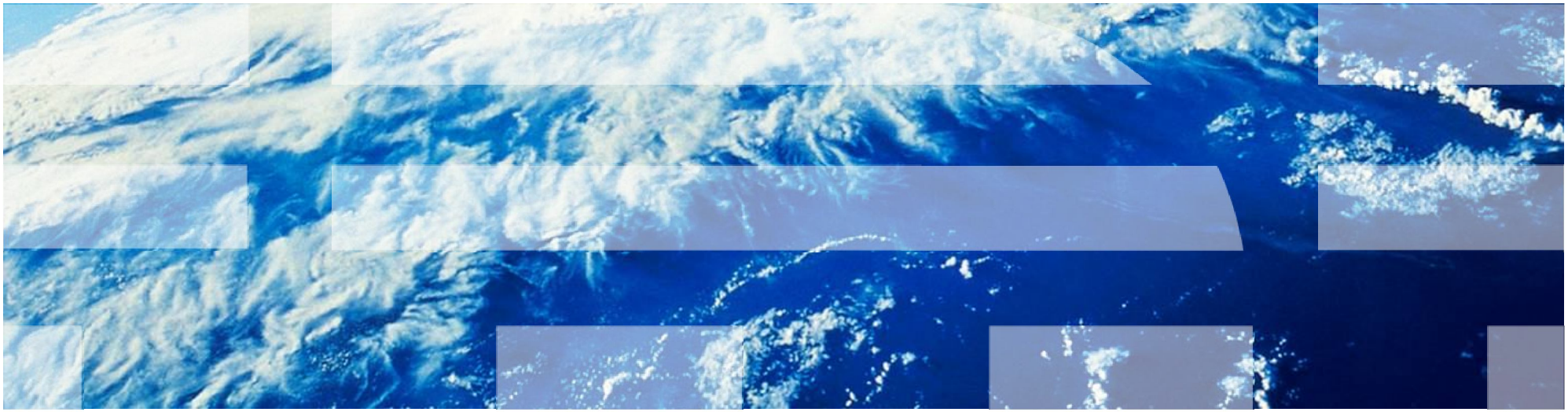
Is It Your deFault?

Removal of BPX.DEFAULT.USER Profile

Laurie Ward CISSP®
z/OS Security Development
IBM Poughkeepsie
LWard@us.ibm.com



Fall 2013

Trademarks

- See url http://www.ibm.com/legal/copytrade.shtml for a list of trademarks.

Summary

What?

z/OS V1.13 is the last release to support FACILITY class profile BPX.DEFAULT.USER

Why?

When BPX.DEFAULT.USER support is used, many users of UNIX System Services can share a UID and GID

What do you need to do?

You must either:

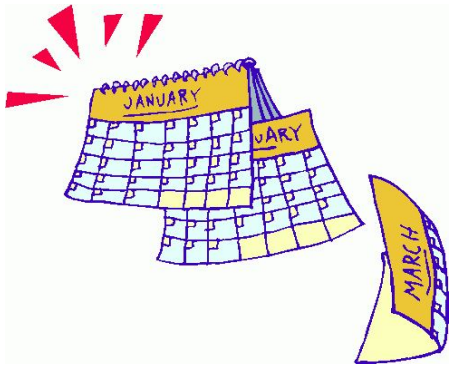
1) Assign a unique UID to each user and GID to each group

-or-

2) Use the BPX.UNIQUE.USER support to **automatically** assign a unique UID to each USS user and a unique GID for their group

A quick history

- MVS Version 4 (1994) OpenEdition MVS support (UNIX)
- OS/390 Release 4 (1997) introduced BPX.DEFAULT.USER profile in FACILITY class
  - A way to allow an MVS user to use UNIX services **without a defined OMVS segment**.
  - Primary purpose was to enable use of UNIX sockets for every FTP user with minimal RACF administration
  - UID defined in the profile could be shared between many users
- z/OS V1R4 (2002) introduced AUTOUID keyword on ALTUSER command
  - Made it easier to generate 'next' unique UID
  - Requires Application Identity Mapping (AIM) Stage 2
- Years passed...
  - IBM encouraged use of unique UIDs assigned to each user
  - More and more Unix services were added
  - Default UIDs were still being used and misused



A quick history...

- z/OS V1R11 (2009) introduced BPX.UNIQUE.USER profile
  - Automatic “on-demand” generation of unique UIDs and GIDs
  - When a z/OS UNIX service is invoked by a user **without an OMVS segment**, a unique UID is permanently assigned
  - Requires Application Identity Mapping (AIM) Stage 3

- z/OS V1.13 (2011) is the last release to support BPX.DEFAULT.USER
  - Statement of Direction from *Preview: z/OS Version 1 Release 13 and z/OS Management Facility Version 1 Release 13 are planned to offer new availability, batch programming, and usability functions*
    - IBM United States Software Announcement 211-007
    - February 15, 2011
    - z/OS V1.13 is planned to be the last release to support BPX.DEFAULT.USER. IBM recommends that you either use the BPX.UNIQUE.USER support that was introduced in z/OS V1.11, or assign unique UIDs to users who need them and assign GIDs for their groups.

A quick history...

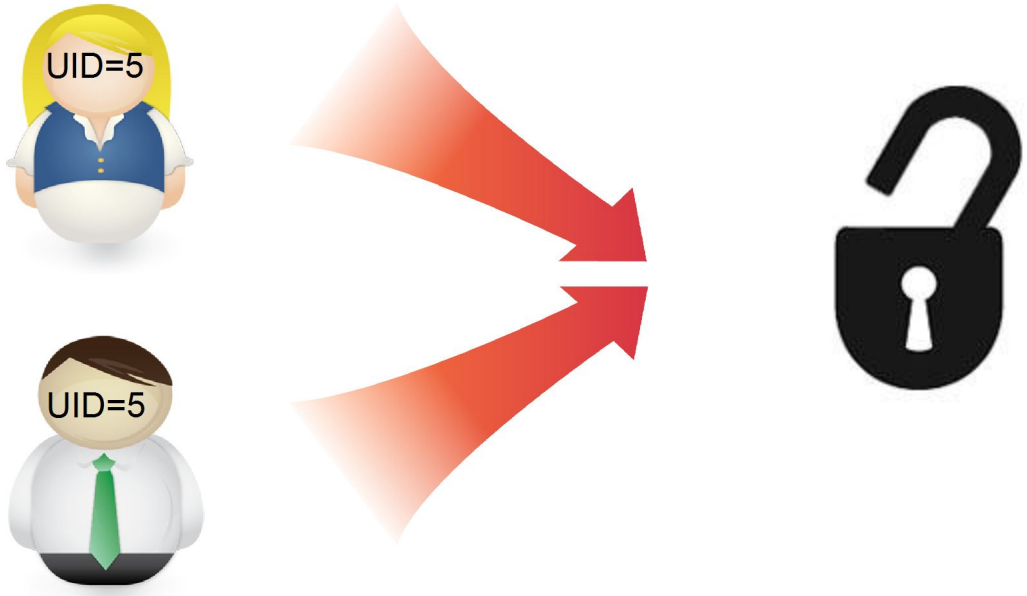
- April 2012 – Health and Migration checks available
  - –APAR OA37164
    - R12: PTF UA64936
    - R13: PTF UA64937

- July 2013 - Support for using &RACUID in OMVS(HOME) field added to ease migration away from BPX.DEFAULT.USER
  - –APAR OA42554 for z/OS V1R12 and V1R13
    - R12: PTF UA69990
    - R13: PTF UA69991

- z/OS V2R1 (September 2013) available with support removed for BPX.UNIQUE.USER profile
  - Yes, support is really gone!
  - You must take action before installing z/OS V2R1!

What's wrong with using BPX.DEFAULT.USER?

- Shared UID produces audit non-conformances
  - No accountability for who did what, who owns what, etc.
- If a Unix service creates a resource while running with a shared UID, that resource is available to all users running with that shared UID
- Certain Unix services are not allowed when user has default UID
  - kill(), sigqueue(), pidaffinity(), ptrace



How do I know if I am using BPX.DEFAULT.USER?

- APAR OA37164 for z/OS V1R12 and V1R13 (and base of V2R1)
  - RACF Health and Migration checks

- Here are some other checks:
  - Does the FACILITY class profile BPX.UNIQUE.USER exist?
    - Yes → then you are not using BPX.DEFAULT.USER
    - No...continue
    - Does the FACILITY class profile BPX.DEFAULT.USER exist?

      Yes → then you are probably using it
  - Check your SMF records
    - Bit which “Indicates a default z/OS UNIX security environment is in effect” is in extended relocate section at location 317 (13D)
    - Event codes 28-58, 60-65
    - SMF unload fields xxxx_DFLT_PROCESS

      xxxx is the prefix to the SMF unload record, such as CMOD, COWN, FACC, IOEP
- RACF downloads page – bpxcheck REXX exec
  - Checks all requirements for using BPX.UNIQUE.USER

How can I stop using BPX.DEFAULT.USER?

OPTION 1: Assign UIDs and GIDs manually

- Use the **UID/GID keywords** to assign a UID to every user and a GID to every group manually
  - ALTUSER MARCY OMVS(UID(859404))
  - ALTGROUP DEPT5 OMVS(GID(354))
- Requirements
  - Must have a procedure to assign a UID/GID for creation of every user/group which will access UNIX resources (new user provisioning)
- Notes
  - Must use this option if your installation has a specific method for assigning UID numbers and GID numbers
  - RACF does not (by default) prevent sharing of UIDs/GIDs

How can I stop using BPX.DEFAULT.USER?

OPTION 2: Assign UIDs and GIDs manually with RACF enforcing uniqueness

- Use the **AUTOUID/AUTOGID keywords** to assign a UID to every user and a GID to every group manually with RACF enforcing uniqueness
  - ALTUSER ANDREW OMVS(AUTOUID))
    - IRR52177I User ANDREW was assigned an OMVS UID value of 4646.
  - ALTGROUP DEPT5 OMVS(AUTOGID))
    - IRR52177I Group DEPT5 was assigned an OMVS GID value of 502.
- Requirements
  - Must have a procedure to assign a UID/GID for creation of every user/group which will access UNIX resources (new user provisioning)
  - RACF database must be at AIM (Application Identity Mapping) stage 2 or 3
  - UNIXPRIV class profile SHARED.IDS must be defined
  - UNIXPRIV class must be active and RACLISTed
  - FACILITY class profile BPX.NEXT.USER must be defined and its APPLDATA field must contain valid ID values or ranges

How can I stop using BPX.DEFAULT.USER?

OPTION 3: Automatically assign UIDs and GIDs as needed

- Use **BPX.UNIQUE.USER profile** to automatically assign a permanent UID to every user and a GID to every group at the time a UNIX service is used
  - Requires no administrative intervention each time a unique ID is assigned
  - Occurs during callable services initUSP, getUMAP, getGMAP
- Requirements:
  - If replacing BPX.DEFAULT.USER, plan alternate access for resources previously accessed through default UID/GID
  - RACF database must be at AIM (Application Identity Mapping) stage 3
  - UNIXPRIV class profile SHARED.IDS must be defined
  - UNIXPRIV class must be active and RACLISTed
  - FACILITY class profile BPX.NEXT.USER must be defined and its APPLDATA field must contain valid ID values or ranges
  - FACILITY profile BPX.UNIQUE.USER must be defined

A look at the Requirements

If replacing BPX.DEFAULT.USER, plan alternate access for resources previously accessed through default UID/GID

- Locate the default UID and GID values in the BPX.DEFAULT.USER profile in the FACILITY class
- Determine which resources the default UID and GID can access
- Authorize the new unique UIDs and GIDs to access the same resources
- Ensure that your plan to maintain UNIX access control lists (ACLs) and GID memberships includes the new unique UIDs and GIDs generated by this method.

A look at the Requirements

RACF database must be at AIM (Application Identity Mapping) stage 2 or 3

- IRRIRA00 utility advances the application identity mapping stage for RACF databases
  - Makes lookup of 'which user maps to UID 45?' faster and more efficient
  - Converts the database mapping profile information into an alias index, which uses less space.
    - Run utility IRRIRA00 with no parameters to check current stage
  - This conversion is accomplished through a series of stage transitions
    - Stage 0 – no alias index, mapping profiles used
    - Stage 1 – alias index created, mapping profiles used
    - Stage 2 – alias index used, mapping profiles maintained
    - Stage 3 – alias index used, mapping profiles deleted
  - If you have more than 129 users sharing a single UID, conversion will fail
    - If you have many, many users sharing UID 0, consider using UNIXPRIV profiles to reduce sharers
    - Can run ICETOOL to check (see *RACF Security Administrator's Guide*)

A look at the Requirements

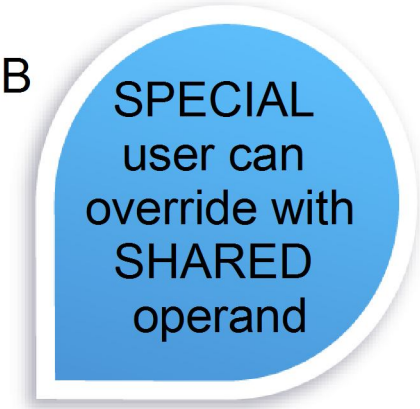
RACF database must be at AIM (Application Identity Mapping) stage 2 or 3...

- Tips on running IRRIRA00
  - Make sure you have run IRRMIN00 PARM=UPDATE for the current release to update the RACF database templates
  - Insure there's space for database growth by running IRRUT200
    - If necessary, run IRRUT400 to increase the amount of space on the RACF database
  - Make a backup copy of RACF database(s) before each IRRIRA00 run
  - Recommend running IRRUT400 after conversion to Stage 1 and Stage 3
  - Converting from Stage 0 to Stage 1 takes the longest – insure CPU time for job is large enough
    - There are no 'in progress' messages...let it run
  - To reduce time the utility runs:
    - Run IRRIRA00 when there is minimal activity on the system
    - Use RVARY to deactivate backup database(s), then run IRRRIA00, then use IRRUT200 or IRRUT400 to copy primary to backup
- See *RACF System Programmer's Guide* for more information

A look at the Requirements

UNIXPRIV class profile SHARED.IDS must be defined

- Acts as a system-wide switch to prevent assignment of an ID which is already in use
  - No generic characters allowed in name: discrete profile name must be used
- Does not affect pre-existing shared IDs
  - Must clean those up separately, if desired
    - SEARCH CLASS(USER) UID(0)
  - Can use IRRDBU00 or IRRICE reports to find shared UIDs and GIDs
    - Samples “UIDS” and “GIDS” in IRRICE member in SYS1.SAMPLIB
- RDEFINE UNIXPRIV SHARED.IDS UACC(NONE)
- SETROPTS CLASSACT(UNIXPRIV) RACLIST(UNIXPRIV)
- Once implemented, it looks like this:
  - ADDUSER MARCY OMVS(UID(12))
    - IRR52174I Incorrect UID 12. This value is already in use by ANDY.
  - ADDGROUP DOGS OMVS(GID(46))
    - IRR52174I Incorrect GID 46. This value is already in use by CATS.



A look at the Requirements

FACILITY class profile BPX.NEXT.USER must be defined and its APPLDATA field must contain valid ID values or ranges

- APPLDATA of BPX.NEXT.USER profile in the FACILITY class is used to derive candidate UID/GID values
  - APPLDATA consists of 2 qualifiers separated by a forward slash ('/')
  - left qualifier specifies starting UID value, or range of UID values
  - right qualifier specifies starting GID value, or range of GID values
  - qualifiers can be null, or specified as 'NOAUTO', to prevent automatic assignment of UIDs or GIDs
- RDEFINE FACILITY BPX.NEXT.USER APPLDATA('data')
  - Some samples for 'data':
    - 1/0
    - 1-50000/1-50000
    - NOAUTO/100000
    - /100000
    - 10000-20000/NOAUTO
    - 10000-20000/

A look at the Requirements

More BPX.NEXT.USER...

- When AUTOUID or AUTOGID is issued, RACF extracts the APPLDATA from BPX.NEXT.USER
  - parses out the starting value
  - checks to see if it is already in use
    - If so, the value is incremented and checked again until an unused value is found
  - assigns the value to the user or group
  - ***replaces*** the APPLDATA with the new starting value
- The administrator can change the APPLDATA at any time using RALTER

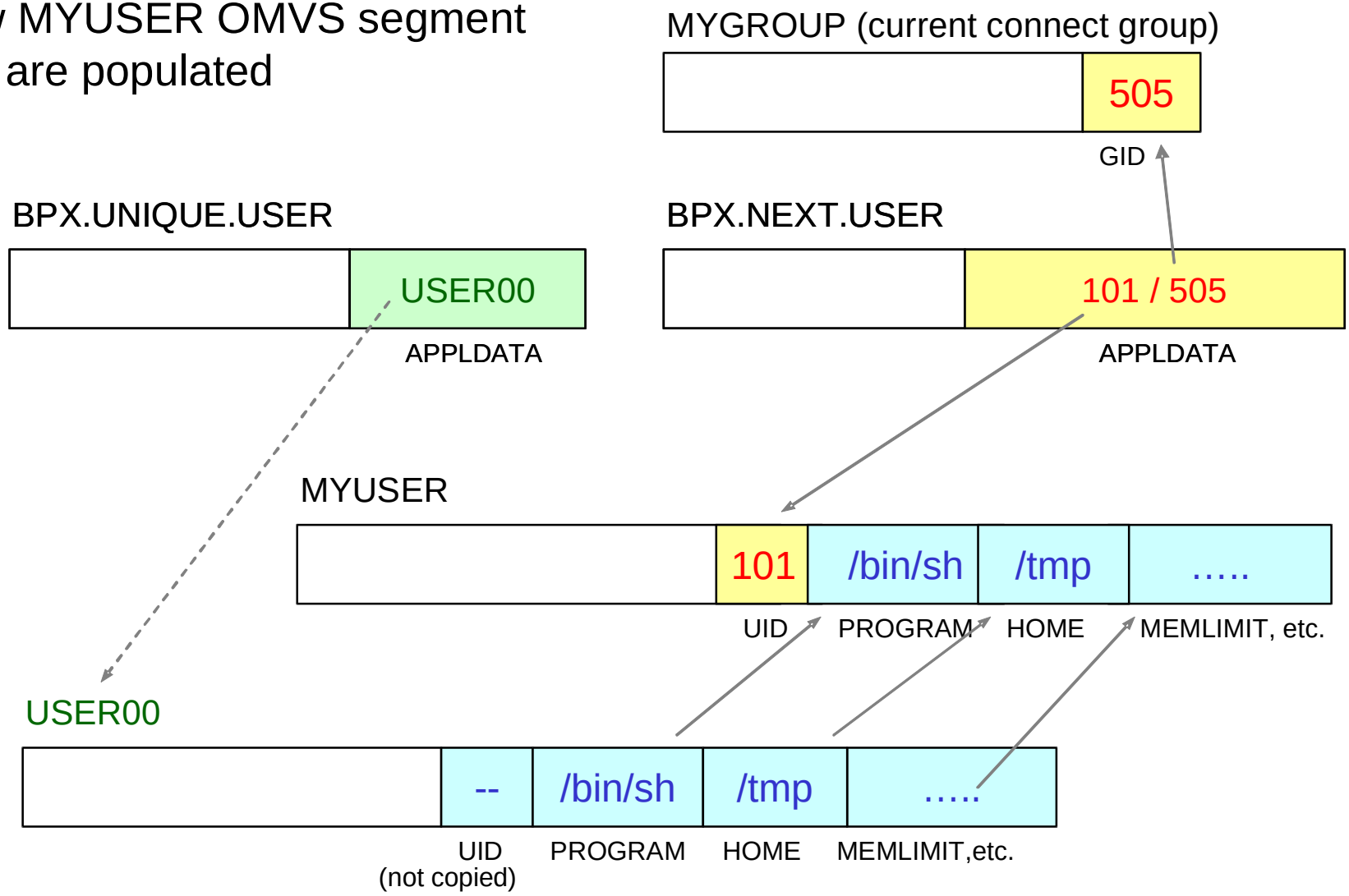
A look at the Requirements

FACILITY profile BPX.UNIQUE.USER must be defined

- Define new FACILITY profile BPX.UNIQUE.USER, and optionally a user profile in APPLDATA field:
  - RDEFINE FACILITY BPX.UNIQUE.USER
    or
  - RDEFINE FACILITY BPX.UNIQUE.USER APPLDATA('USER00')
- After this profile is created
  - then BPX.DEFAULT.USER is not considered.
  - For a user or group without an OMVS segment, a unique UID or GID is assigned when UNIX service is used
    - Unique UID/GID is derived from BPX.NEXT.USER profile just as for AUTOUID/AUTOGID keywords on ALTUSER/ALTGROUP
    - Unique UID/GID is permanently stored in OMVS segment automatically
  - If a user name is specified in APPLDATA, its other OMVS fields are copied to the target user when the new UID is saved

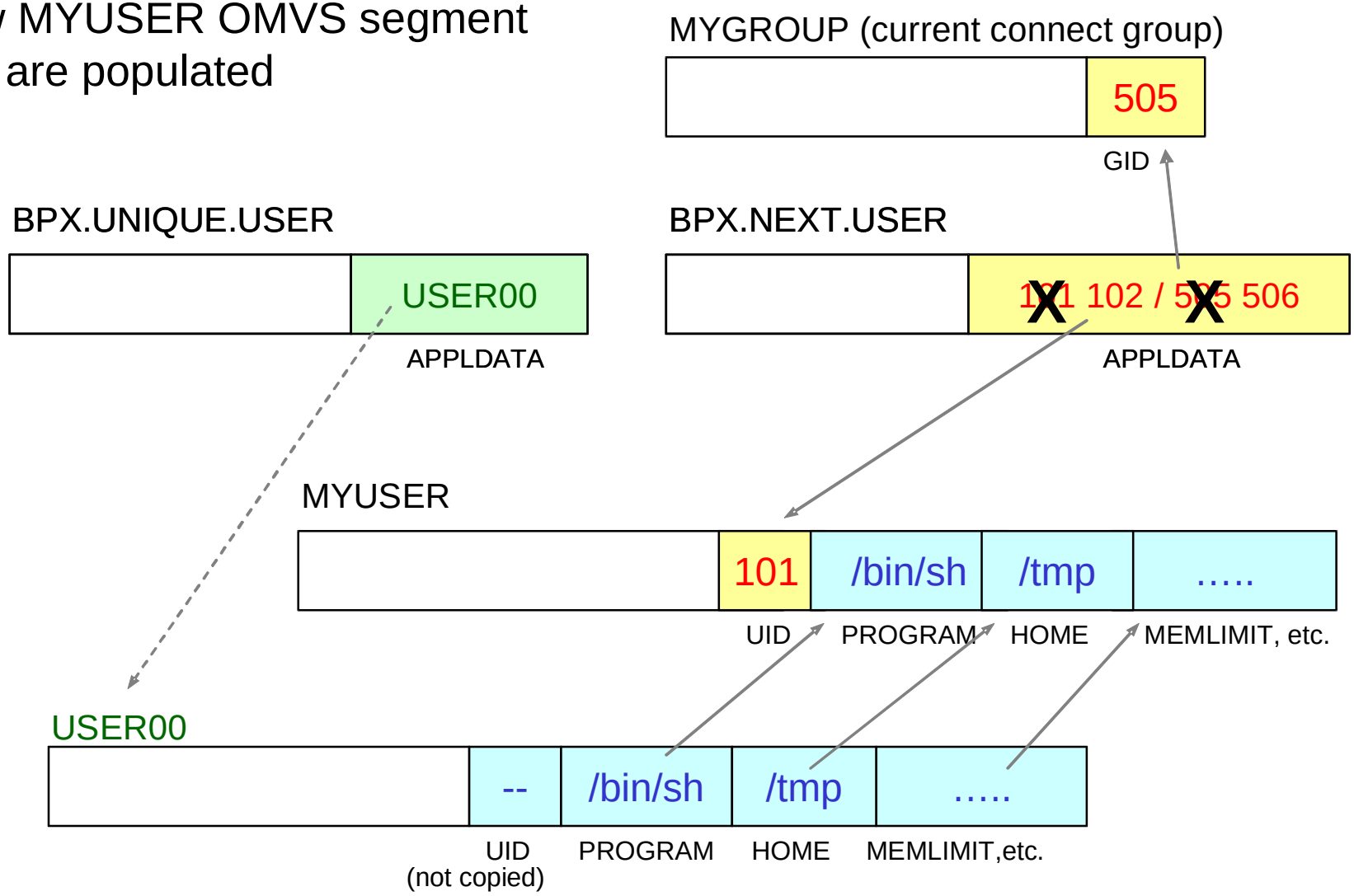
The view in pictures when BPX.UNIQUE.USER is used

- How MYUSER OMVS segment fields are populated



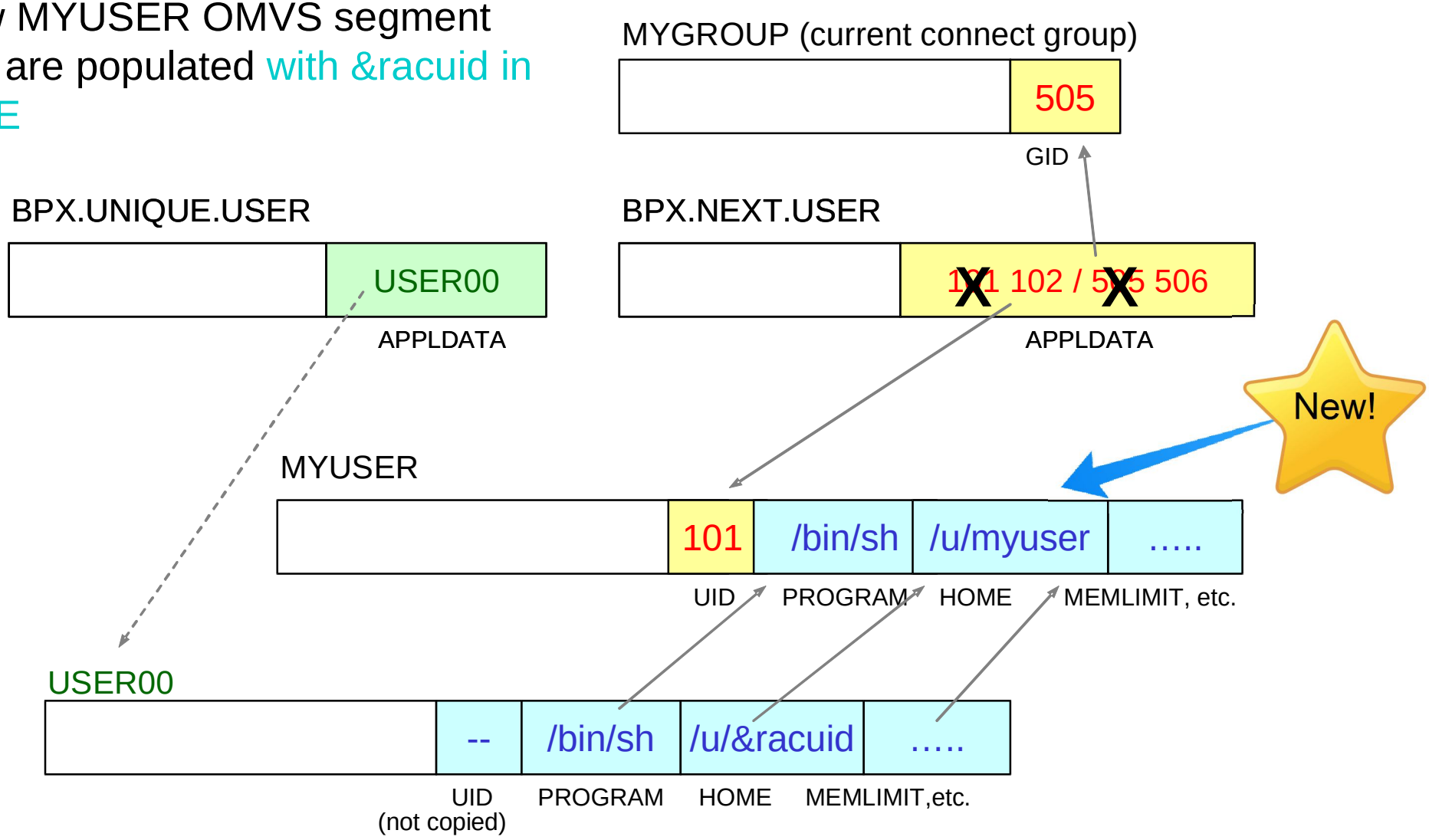
The view in pictures when BPX.UNIQUE.USER is used...

- How MYUSER OMVS segment fields are populated



The view in pictures when BPX.UNIQUE.USER is used...

- How MYUSER OMVS segment fields are populated with &racuid in HOME



How does new &RACUID support work?

- Enhancement to BPX.UNIQUE.USER to allow specification of &RACUID in the home directory field of the model user's OMVS segment.
  - ALTUSER USER00 OMVS(HOME(/u/&racuid))
- Substitutes user ID for &racuid when a new OMVS segment is created for a user using BPX.UNIQUE.USER
- In upper case if “&RACUID” is specified
- In lower case if any lower case characters are specified, like “&Racuid”
- When using automount, this *eliminates all manual intervention*
  - a user file system is allocated, mounted, and assigned the user ID as its owner
- Notes
  - Only the first occurrence of &racuid is substituted
  - If the substitution would result in a path name exceeding 1023 characters (the max), then substitution is not performed.
  - If sharing the RACF database with a downlevel system, substitution will not be performed on the downlevel system

How does new &RACUID support work?....

- Example:
  - ADDUSER USER00 NAME('OMVS model user profile')
    OMVS(HOME('/u/&racuid') PROGRAM('/bin/sh'))
    NOPASSWORD RESTRICTED

  - RDEFINE FACILITY BPX.UNIQUE.USER APPLDATA('USER00')

  - TSO user LAURIE has no OMVS segment, logs onto TSO, enters 'ishell'
    - RACF defines an OMVS segment for user profile LAURIE with a UID assigned from BPX.NEXT.USER profile, PROGRAM('/bin/sh'), and HOME('/u/laurie')
    - If default group PROGMR for LAURIE does not have an OMVS segment, RACF defines an OMVS segment for group PROGMR with a GID assigned from BPX.NEXT.USER profile
    - UNIX System Services allocates and mounts directory /u/laurie

- APAR OA42554 for z/OS V1R12 and V1R13

Recommendations

- Convert RACF database to AIM Stage 3
- Define UNIXPRIV profile SHARED.IDS
- Activate and RACLIST the UNIXPRIV class
- Define FACILITY profile BPX.NEXT.USER
  - Set APPLDATA to point to a model user
  - Use &racuid in the HOME field for the model user
- Define FACILITY profile BPX.UNIQUE.USER

What happens if I do nothing?

- In z/OS V1R13, nothing changes
  - You may get warning messages from the z/OS Health Checker
- In z/OS V2R1, BPX.DEFAULT.USER profile will be ignored
  - You may get warning messages from the z/OS Health Checker
  - Users with no OMVS segment or no UID will not be able to run any Unix service

Summary

What?

z/OS V1.13 is the last release to support
FACILITY class profile
BPX.DEFAULT.USER

What do you need to do?

You must either:

1) Assign a unique UID to each user and GID to each group

-or-

2) Use the BPX.UNIQUE.USER support to **automatically** assign a unique UID to each USS user and a unique GID for their group

Any Questions?

Helpful Publications

- SA22-7691 - z/OS Security Server RACF Callable Services
- SA22-7687 - z/OS Security Server RACF Command Language Reference
- GA22-7680 - z/OS Security Server RACF Data Areas
- SA22-7682 - z/OS Security Server RACF Macros and Interfaces
- SA22-7686 - z/OS Security Server RACF Messages and Codes
- SA22-7683 - z/OS Security Server RACF Security Administrator's Guide
- SA22-7681 - z/OS Security Server RACF System Programmer's Guide
- SA22-7692 - z/OS Security Server RACROUTE Macro Reference
- GA22-7689 - z/OS Security Server RACF Diagnosis Guide
- GA22-7800 – z/OS UNIX System Services Planning

Helpful Websites

- RACF downloads
  - http://www.ibm.com/systems/z/os/zos/features/racf/goodies.html
- RACF resources - presentations, user groups, education
  - http://www-03.ibm.com/systems/z/os/zos/features/racf/resources.html