Rocket

# Rocket LDAP Bridge

Jared Hunter | June 20, 2014

Jared Hunter
Managing Director of R&D, Security Products

jhunter@rocketsoftware.com

# Overview

- What is the Rocket LDAP Bridge?
  - Architecture, components, installation
- What can it do?
  - Functionality of various plugins
- A few usage examples

# What is the LDAP Bridge?

# LDAP Bridge Architecture Basics

- Runs on all IBM-supported versions of z/OS
- Is itself an LDAP Server
  - Based on a Rocket port of OpenLDAP
  - Accepts LDAP protocol connections over TCP/IP, supports SSL/TLS connections (including mutual authentication)
  - Modified to support our plugin architecture
- Most interesting functionality is in the plugins

# LDAP Bridge Components

- Server and plugin shared library code, scripts
  - Installed in UNIX System Services, logs are written to the HFS
  - Primarily implemented in C++ and HLASM
- System exit installed in MVS
  - Allows ongoing changes made natively in the ESM (e.g. RACF) to be picked up and modeled in the LDAP database
  - Primarily implemented in HLASM
- Server started task launch JCL
  - Copied to MVS location as part of installation script

# LDAP Bridge Installation / Configuration

- Product archives initially uploaded and unpacked in USS
- Interactive shell script prompts for installation details, constructs configuration files and JCL, creates MVS data sets
- Some manual steps
  - APF authorization, Activating and verifying SMF installation exit, Setting RACF option to generate SMF records

# Installation / Configuration (2 of 2)

- Install plugins
- Initial load of the LDAP database
  - Extract existing security database contents and load them into the LDAP Bridge database
- Run as a submitted job, Verify installation with provided utilities
- Run as a started task

# What can the LDAP Bridge do?

# Rocket LDAP Bridge functionality

- It can do a great many things!
  - …but these are specific actions that occur on command, when commanded
  - No embedded business logic
  - No built-in automation (aside from keeping its database up to date)

# LDAP Bridge functionality

- After loading RACF data into LDAP database
  - ldap_bind to authenticate with RACF credentials
  - ldap_search to find and read mainframe security information about users, groups, data sets, and resource profiles
- Everything else uses plugins
  - Keeping LDAP contents up to date in real-time via SMF exit
  - Changing RACF in response to LDAP add, modify, delete
  - Executing TSO commands
  - Generating PassTickets

# LDAP Bridge plugins

- racf2ldap (and acf22ldap, tss2ldap)
  - Read SMF record data (as collected by the SMF Installation Exit), and modify LDAP database accordingly
- ldap2racf (and ldap2acf2, ldap2tss)
  - Translate LDAP changes into commands to the ESM
- ldap2tso
  - Issue non-interactive TSO commands in the security context of the logged-in user
  - Commonly used to create catalog alias for new TSO users

# LDAP Bridge plugins (2 of 2)

- pwdsync
  - Adds support for exposing read access to user passwords via LDAP directory
- pticket
  - Generate and return a RACF PassTicket credential to a sufficiently authorized LDAP client
  - Used to enable single sign-on functionality in web applications
- ldifsync
  - Retrieve a bulk representation (in LDIF format) of the database changes that occurred since the last ldifsync query

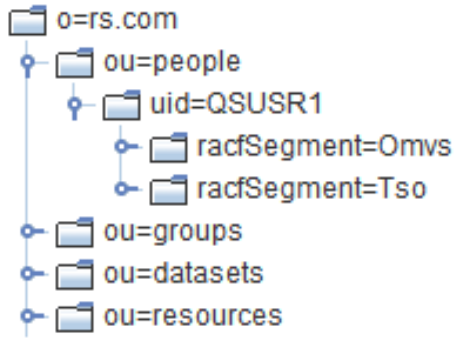
# Usage Examples

# Example conventions

- Instead of showing LDAP API calls, we'll use LDIF format to describe operations
- LDAP Browser screens or RACF command output will show results
- In very limited cases, output has been edited to redact Rocket-internal system information

# A note about LDAP schema

- The test system's o=rs.com Root DN, as viewed in an LDAP browser:




- Note that profile types (Users, Groups, Data Set profiles, Resource profiles) are split into four top-level OrgUnits

# Examples using the ldap2racf plugin

- User management
  - Adding a user, Changing a user property
- Group management
  - Creating a group, Editing group membership
- Data set profile management
- Resource profile management

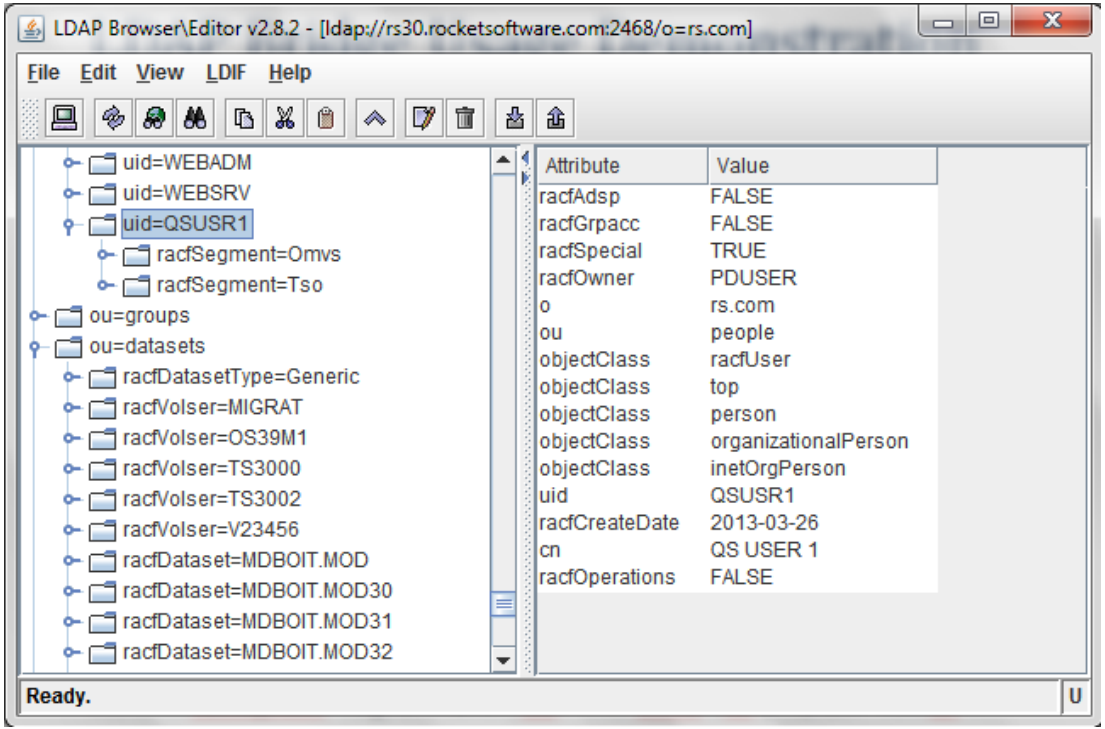
# Adding a User – LDIF content (1 of 2)

```
dn: uid=QSUSR1, ou=people, o=rs.com
racfAdsp: FALSE
racfGrpacc: FALSE
racfSpecial: TRUE
racfOwner: PDUSER
racfRevoke: FALSE
mail: qsusr1@rs.com
uid: QSUSR1
objectClass: inetOrgPerson
objectClass: organizationalPerson
objectClass: person
objectClass: racfUser
objectClass: top
racfCreateDate: 2013-03-20
cn: QS User 1
racfOperations: FALSE
```

# Adding a User – LDIF content (2 of 2)

```
dn: racfSegment=Omvs, uid=QSUSR1, ou=people, o=rs.com
racfOmvsUid: 0000001600
racfOmvsHome: /u/qsusr1
racfOmvsFileprocmax: 065535
uid: QSUSR1
objectClass: racfUserSegOmvs
objectClass: top
racfSegment: Omvs
racfOmvsProgram: /bin/tcsh

dn: racfSegment=Tso, uid=QSUSR1, ou=people, o=rs.com
racfTsoProc: ROCKPROC
racfTsoMaxsize: 0060000
uid: QSUSR1
racfTsoSize: 0006000
objectClass: racfUserSegTso
objectClass: top
racfTsoAcctnum: ACCT#
racfSegment: Tso
```

# Adding a User – confirming LDAP change



# Adding a User – Generated RACF commands

```
ADDUSER QSUSR1 NOADSP NOGRPACC SPECIAL OWNER(PDUSER)
      NAME('QS User 1') NOOPERATIONS

ALTUSER QSUSR1 OMVS(UID(0000001600) FILEPROCMAX(065535)
      HOME('/u/qsusr1') PROGRAM('/bin/tcsh'))

ALTUSER QSUSR1 TSO(MAXSIZE(0060000) PROC(ROCKPROC) SIZE(0006000)
      ACCTNUM('ACCT#'))
```

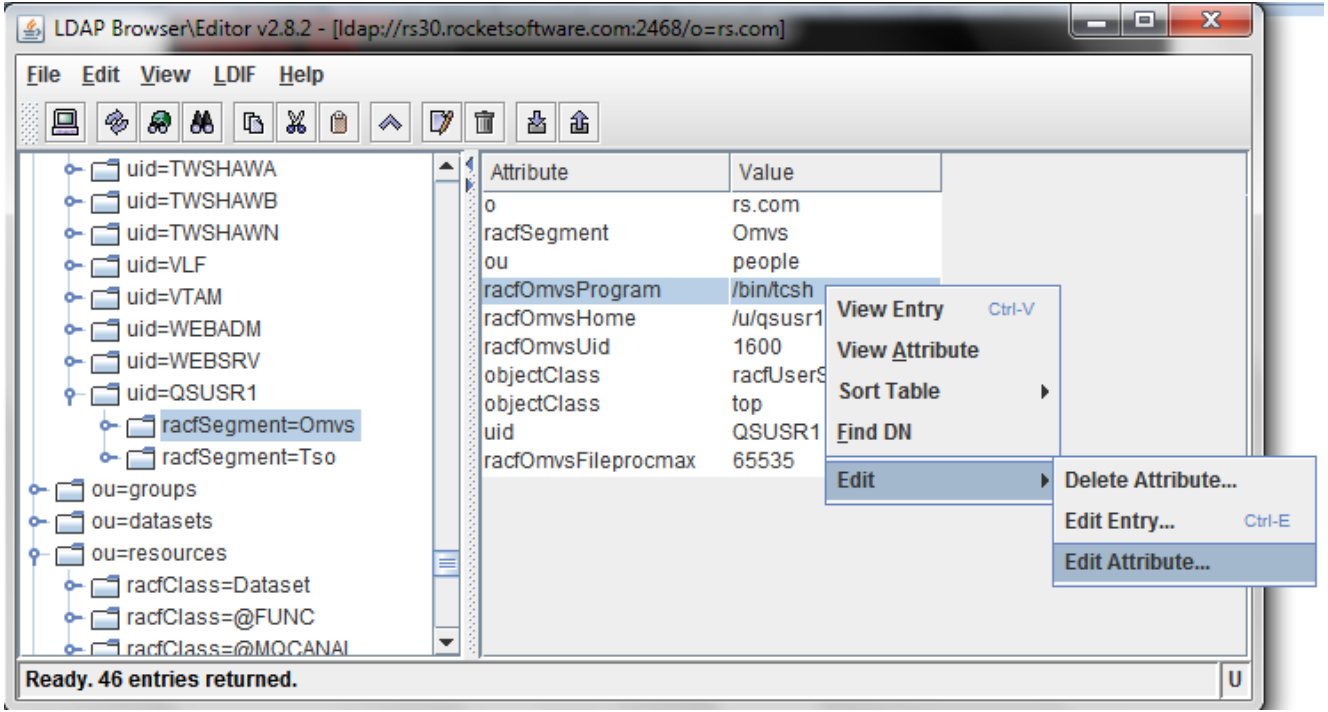
# Adding a User – confirming RACF change

```
LISTUSER QSUSR1
 USER=QSUSR1   NAME=QS USER 1              OWNER=PDUSER    CREATED=13.084
  DEFAULT-GROUP=PDUSER    PASSDATE=00.000 PASS-INTERVAL=180 PHRASEDATE=N/A
  PASSWORD ENVELOPED=NO
  ATTRIBUTES=SPECIAL
  REVOKE DATE=NONE    RESUME DATE=NONE
  LAST-ACCESS=UNKNOWN
  CLASS AUTHORIZATIONS=NONE
  NO-INSTALLATION-DATA
  NO-MODEL-NAME
  LOGON ALLOWED    (DAYS)          (TIME)
  -----------------------------------------
  ANYDAY                           ANYTIME
   GROUP=PDUSER     AUTH=USE      CONNECT-OWNER=PDUSER    CONNECT-DATE=13.084
     CONNECTS=     00  UACC=NONE      LAST-CONNECT=UNKNOWN
     CONNECT ATTRIBUTES=NONE
     REVOKE DATE=NONE    RESUME DATE=NONE
 SECURITY-LEVEL=NONE SPECIFIED
 CATEGORY-AUTHORIZATION
  NONE SPECIFIED
 SECURITY-LABEL=NONE SPECIFIED
 READY
```

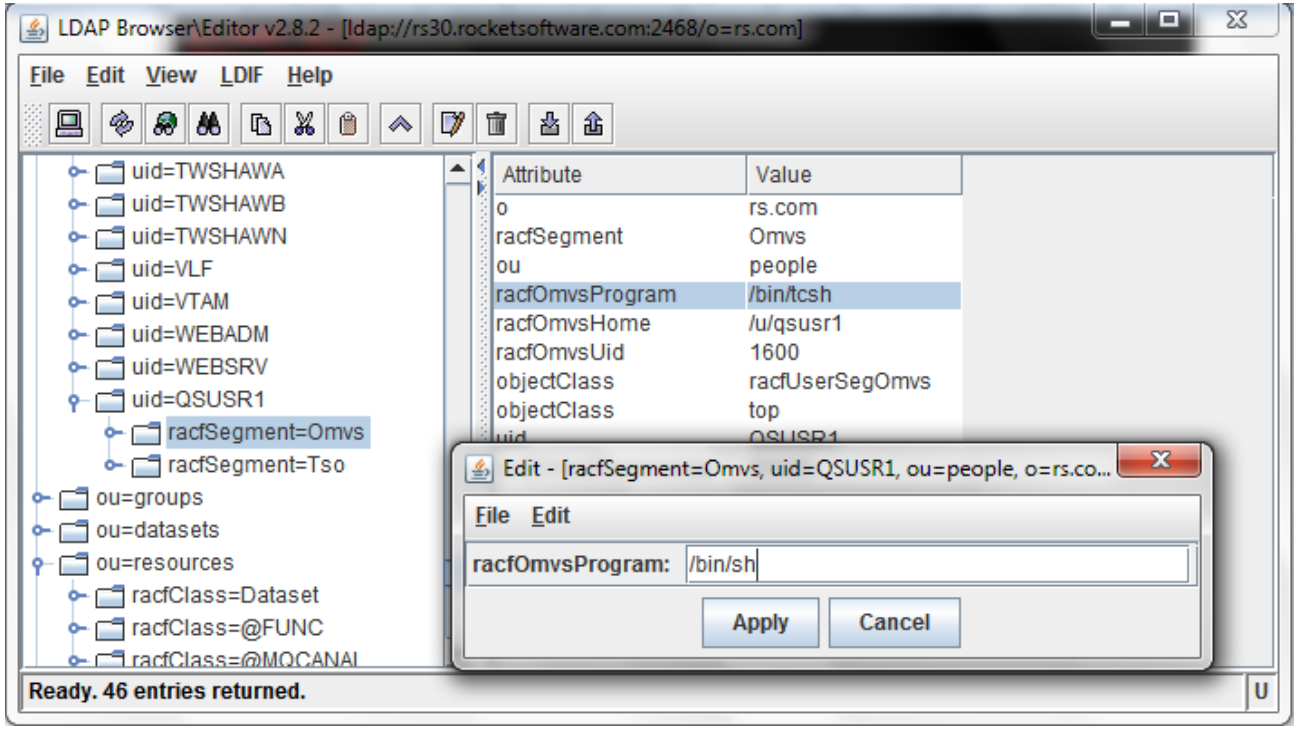
# Changing a User Property

- We'll change a User's OMVS login shell using an LDAP Browser
  - Modify the racfOmvsProgram attribute of the QSUSR1 user
  - Changing the shell from /bin/tcsh to /bin/sh

# Editing a User via an LDAP Browser (1 of 2)



# Editing a User via an LDAP Browser (2 of 2)



# Editing a User – LDAP Server log entry

2013-03-26-04.30.06.749884: conn=3 op=6 ldap2esm: IssueCommand: executing command under PDXIA authorization:
2013-03-26-04.30.06.749911: ALTUSER QSUSR1 OMVS(PROGRAM('/bin/sh'))
2013-03-26-04.30.06.795721: conn=3 op=6 ldap2esm: IssueCommand: IRRSEQ00 SAF return = 0, return = 0, reason = 0
2013-03-26-04.30.06.795805: conn=3 op=6 RESULT tag=103 err=0 text=ALTUSER QSUSR1 OMVS(PROGRAM('/bin/sh'))
2013-03-26-04.30.06.795814: (no message)
2013-03-26-04.30.06.948176: conn=0 op=26 MOD dn="racfSegment=Omvs,uid=QSUSR1,ou=people,o=rs.com"
2013-03-26-04.30.06.948225: conn=0 op=26 MOD attr=racfOmvsProgram
2013-03-26-04.30.06.964779: conn=0 op=26 RESULT tag=103 err=0 text=

Snippet of slapd.log (located at *installationDirectory*/logs/slapd.log)

# Editing a User – confirming RACF change

```
LISTUSER QSUSR1 OMVS NORACF
 USER=QSUSR1

 OMVS INFORMATION
 ----------------
 UID= 0000001600
 HOME= /u/qsusr1
 PROGRAM= /bin/sh
 CPUTIMEMAX= NONE
 ASSIZEMAX= NONE
 FILEPROCMAX= 00065535
 PROCUSERMAX= NONE
 THREADSMAX= NONE
 MMAPAREAMAX= NONE

 READY
```

# Creating a Group – LDIF content

```
dn: cn=QSGRP1, ou=groups, o=rs.com
racfOwner: SYS1
racfTermuacc: TRUE
racfCreateDate: 2013-03-26
racfUacc: NONE
objectClass: groupOfNames
objectClass: racfGroup
objectClass: top
racfUniversal: FALSE
racfSupgroup: SYS1
cn: QSGRP1

dn: racfSegment=Omvs, cn=QSGRP1, ou=groups, o=rs.com
racfOmvsGid: 0000003000
objectClass: racfGroupSegOmvs
objectClass: top
racfSegment: Omvs
cn: QSGRP1
```

# Creating a Group – LDAP Server log entry

2013-03-26-05.22.41.402396: conn=6 op=3 ldap2esm: IssueCommand: executing command under PDXIA authorization:
2013-03-26-05.22.41.402415: ADDGROUP QSGRP1 OWNER(SYS1) TERMUACC SUPGROUP(SYS1)
2013-03-26-05.22.41.533827: conn=6 op=3 ldap2esm: IssueCommand: IRRSEQ00 SAF return = 0, return = 0, reason = 0
2013-03-26-05.22.41.533931: conn=6 op=3 RESULT tag=105 err=0 text=ADDGROUP QSGRP1 OWNER(SYS1) TERMUACC SUPGROUP(SYS1)
2013-03-26-05.22.41.533940: (no message)

2013-03-26-05.22.41.832586: conn=6 op=4 ADD dn="racfSegment=Omvs,cn=QSGRP1,ou=groups,o=rs.com"
2013-03-26-05.22.41.833294: conn=6 op=4 ldap2esm: IssueCommand: executing command under PDXIA authorization:
2013-03-26-05.22.41.833320: ALTGROUP QSGRP1 OMVS(GID(0000003000))
2013-03-26-05.22.41.879912: conn=6 op=4 ldap2esm: IssueCommand: IRRSEQ00 SAF return = 0, return = 0, reason = 0
2013-03-26-05.22.41.879938: conn=6 op=4 RESULT tag=105 err=0 text=ALTGROUP QSGRP1 OMVS(GID(0000003000))
2013-03-26-05.22.41.879946: (no message)

Snippet of slapd.log (located at *installationDirectory*/logs/slapd.log)

# Creating a Group – confirming RACF change

```
LISTGRP QSGRP1 OMVS
 INFORMATION FOR GROUP QSGRP1
     SUPERIOR GROUP=SYS1          OWNER=SYS1          CREATED=13.085
     NO INSTALLATION DATA
     NO MODEL DATA SET
     TERMUACC
     NO SUBGROUPS
     NO USERS

 OMVS INFORMATION
 ----------------
 GID= 0000003000
 READY
```

# Group Membership – LDIF content

- The LDAP group object uses the ‘member’ attribute to map users belonging to the group
- Two user IDs in this group:

```
dn: cn=DEVUSS, ou=groups, o=rs.com
racfOwner: SYS1
racfSupgroup: SYS1
racfUniversal: FALSE
racfUacc: NONE
member: uid=MDDEJPH,ou=people,o=rs.com
member: uid=MDTEJPH,ou=people,o=rs.com
objectClass: groupOfNames
objectClass: racfGroup
objectClass: top
racfTermuacc: TRUE
racfCreateDate: 2007-10-15
cn: DEVUSS
racfData: USS ACCESS FOR DEVELOPMENT
```

# Adding a User to a Group – LDIF content

- To add a user to a group, we modify a group object to add a member attribute:

```
dn: cn=QSGRP1,ou=groups,o=rs.com
changetype: modify
add: member
member: uid=QSUSR1,ou=people,o=rs.com
```

# Adding a User to a Group – LDAP Server log

2013-03-26-08.58.41.659291: conn=19 op=1 MOD attr=member
2013-03-26-08.58.41.666803: conn=19 op=1 ldap2esm: IssueCommand: executing command under PDXIA authorization:
2013-03-26-08.58.41.666826: CONNECT QSUSR1 GROUP(QSGRP1)
2013-03-26-08.58.41.831763: conn=19 op=1 ldap2esm: IssueCommand: IRRSEQ00 SAF return = 0, return = 0, reason = 0
2013-03-26-08.58.41.831811: conn=19 op=1 RESULT tag=103 err=0 text=CONNECT QSUSR1 GROUP(QSGRP1)
2013-03-26-08.58.41.831819: (no message)

Snippet of slapd.log (located at *installationDirectory*/logs/slapd.log)

# Adding a User to a Group – confirming RACF change

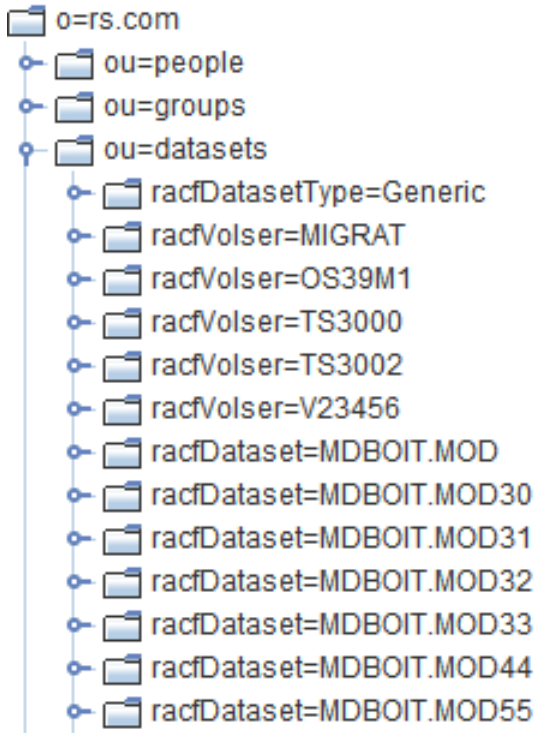
```
LISTGRP QSGRP1
 INFORMATION FOR GROUP QSGRP1
     SUPERIOR GROUP=SYS1          OWNER=SYS1          CREATED=13.085
     NO INSTALLATION DATA
     NO MODEL DATA SET
     TERMUACC
     NO SUBGROUPS
     USER(S)=      ACCESS=      ACCESS COUNT=      UNIVERSAL ACCESS=
       QSUSR1        USE            000000               NONE
           CONNECT ATTRIBUTES=NONE
           REVOKE DATE=NONE                  RESUME DATE=NONE
 READY
```

# Data set profile management

- Discrete data set profiles
  - Cataloged vs. Uncataloged
  - Profiles for data sets that aren't catalogued are stored under an LDAP sub-tree that includes the VOLSER
    - Cataloged: dn: racfDataset=PDCN04.SAMPMOD, ou=datasets, o=rs.com
    - Uncataloged: dn: racfDataset=TSTUSER.HCD.MSGLOG, racfVolser=TS3000, ou=datasets, o=rs.com
- Generic data set profiles are also supported

dn: racfDataset=ADSM.*.**, racfDatasetType=Generic, ou=datasets, o=rs.com

# Data set profile management – schema hierarchy



# Add a generic data set profile – LDIF (1 of 2)

- In this example, we'll create a Generic profile to cover some of a user's data sets, and give that user ALTER authority to the new generic profile

```
dn: racfDataset=PDXIA.SAMPLE.**, racfDatasetType=Generic, ou=datasets, o=rs.com
o: rs.com
racfDatasetType: GENERIC
racfClass: DATASET
ou: datasets
racfUacc: NONE
objectClass: racfDataset
objectClass: top
racfDataset: PDXIA.SAMPLE.**
```

# Add a generic data set profile – LDIF (2 of 2)

- Granting the user access to the new generic profile:

```
dn: racfPermitId=PDXIA, racfDataset=PDXIA.SAMPLE.**, racfDatasetType=Generic,
  ou=datasets, o=rs.com
racfAccess: ALTER
racfPermitId: PDXIA
ou: datasets
objectClass: racfDatasetPermitId
objectClass: top
racfDatasetType: GENERIC
racfDataset: PDXIA.SAMPLE.**
o: rs.com
```

# Generic data set profile – LDAP Server log

2013-03-27-01.42.50.404410: conn=31 op=77 ldap2esm: IssueCommand: executing command under PDXIA authorization:
2013-03-27-01.42.50.404436: ADDSD 'PDXIA.SAMPLE.**' Generic AUDIT( FAILURES(READ)) UACC(NONE)
2013-03-27-01.42.50.435581: conn=31 op=77 ldap2esm: IssueCommand: IRRSEQ00 SAF return = 0, return = 0, reason = 0
2013-03-27-01.42.50.435662: conn=31 op=77 RESULT tag=105 err=0 text=ADDSD 'PDXIA.SAMPLE.**' Generic AUDIT( FAILURES(READ)) U
CC(NONE)
2013-03-27-01.42.50.435672: (no message)
2013-03-27-01.42.50.758803: conn=31 op=78 ADD dn="racfPermitId=PDXIA,racfDataset=PDXIA.SAMPLE.**,racfDatasetType=Generic,ou=
atasets,o=rs.com”
2013-03-27-01.42.50.772764: conn=31 op=78 ldap2esm: IssueCommand: executing command under PDXIA authorization:
2013-03-27-01.42.50.772774: PERMIT 'PDXIA.SAMPLE.**' ID(PDXIA) Generic ACCESS(ALTER)
2013-03-27-01.42.50.787412: conn=31 op=78 ldap2esm: IssueCommand: IRRSEQ00 SAF return = 0, return = 0, reason = 0
2013-03-27-01.42.50.787486: conn=31 op=78 RESULT tag=105 err=0 text=PERMIT 'PDXIA.SAMPLE.**' ID(PDXIA) Generic ACCESS(ALTER)
2013-03-27-01.42.50.787495: (no message)

# Generic data set profile – confirming RACF change

```
LISTDSD DATASET('PDXIA.SAMPLE.** ') ALL

INFORMATION FOR DATASET PDXIA.SAMPLE.** (G)

LEVEL  OWNER    UNIVERSAL ACCESS   WARNING   ERASE
-----  -------- ----------------   -------   -----
 00    PDXIA          NONE          NO       NO

AUDITING
--------
FAILURES(READ)

[ . . . Content stripped to fit on this slide . . . ]

  ID     ACCESS
--------  -------
PDXIA      ALTER

  ID    ACCESS   CLASS              ENTITY NAME
-------- ------- -------- ----------------------------------------------------
NO ENTRIES IN CONDITIONAL ACCESS LIST
READY
```

# Resource profile management

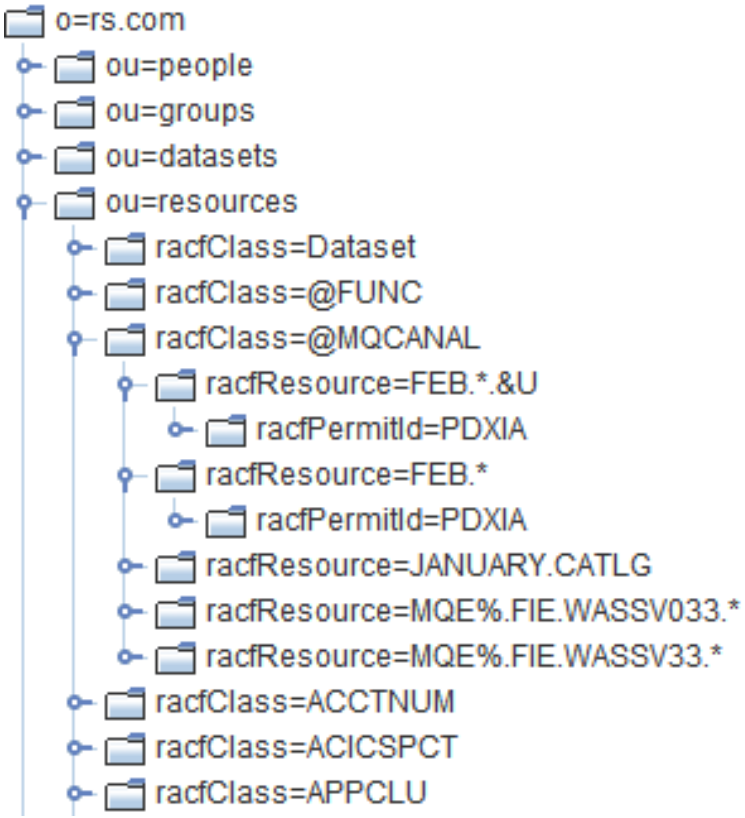
- For each resource class, there is a corresponding LDAP entry under the ou=resources sub-tree
  - For resource class TSOPROC, the LDIF is:

```
dn: racfClass=TSOPROC, ou=resources, o=rs.com
objectClass: racfClass
objectClass: top
racfClass: TSOPRO
```

- RACF resource profiles are under the subtree of the resource class
  - The DN of BBOPROC in class TSOPROC is:

```
dn: racfResource=BBOPROC, racfClass=TSOPROC, ou=resources, o=rs.com
```

# Resource profile management – schema hierarchy



# Add a resource profile permission - LDIF

- In this example, user SYSADM is granted READ authority to BBOPROC in class TSOPROC

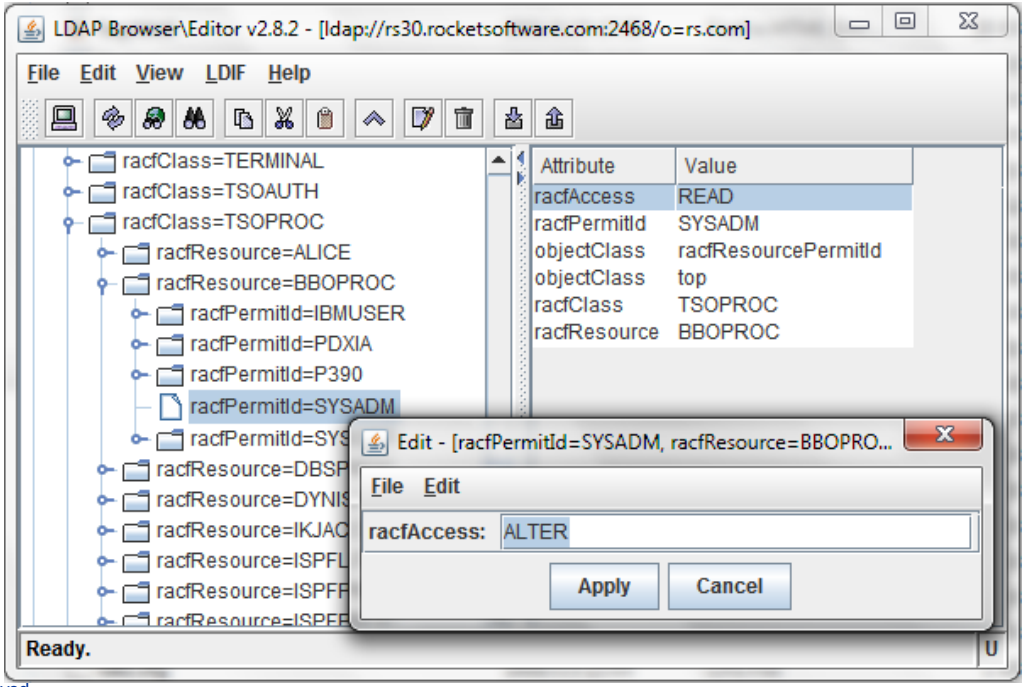
```
dn: racfPermitId=SYSADM, racfResource=BBOPROC, racfClass=TSOPROC,
    ou=resources, o=rs.com
racfAccess: READ
racfPermitId: SYSADM
objectClass: racfResourcePermitId
objectClass: top
racfClass: TSOPROC
racfResource: BBOPROC
```

# Add resource permission – confirming RACF change

```
------
NO USER TO BE NOTIFIED

CREATION DATE  LAST REFERENCE DATE  LAST CHANGE DATE
 (DAY) (YEAR)       (DAY) (YEAR)      (DAY) (YEAR)
-------------  -------------------  ----------------
  345    01          345    01         345    01

ALTER COUNT   CONTROL COUNT   UPDATE COUNT   READ COUNT
-----------   -------------   ------------   ----------
  000000         000000         000000         000000

USER      ACCESS   ACCESS COUNT
----      ------   ------ -----
R390      READ          000000
SYSADM    READ          000000
SYSOPR    READ          000000
IBMUSER   ALTER         000000
PDXIA     ALTER         000000

   ID     ACCESS  ACCESS COUNT  CLASS                    ENTITY  NAME
-------- -------  ------------ -------- ----------------------------------------
NO ENTRIES IN CONDITIONAL ACCESS LIST
READY
```

# Changing a resource profile permission

- In this example, we'll use an LDAP Browser to change the permission we just granted from READ to ALTER



# Changing a permission – LDAP Server log

2013-03-27-02.57.41.094281: conn=35 op=10 SEARCH RESULT tag=101 err=0 nentries=1 text=
2013-03-27-02.58.50.598423: conn=35 op=11 MOD
dn="racfPermitId=SYSADM,racfResource=BBOPROC,racfClass=TSOPROC,ou=resources,o=
s.com"
2013-03-27-02.58.50.598505: conn=35 op=11 MOD attr=racfAccess
2013-03-27-02.58.50.623386: conn=35 op=11 ldap2esm: IssueCommand: executing command under PDXIA
authorization:
2013-03-27-02.58.50.623412: PERMIT BBOPROC CLASS(TSOPROC) ID(SYSADM) ACCESS(ALTER)
2013-03-27-02.58.50.692423: conn=35 op=11 ldap2esm: IssueCommand: IRRSEQ00 SAF return = 0, return =
0, reason = 0
2013-03-27-02.58.50.705528: conn=35 op=11 RESULT tag=103 err=0 text=PERMIT BBOPROC
CLASS(TSOPROC) ID(SYSADM) ACCESS(ALTER)

# Changing resource permission – confirming in RACF

```
CREATION DATE  LAST REFERENCE DATE  LAST CHANGE DATE
 (DAY) (YEAR)       (DAY) (YEAR)      (DAY) (YEAR)
-------------  -------------------  ----------------
  345    01          345    01          345    01

ALTER COUNT   CONTROL COUNT   UPDATE COUNT   READ COUNT
-----------   -------------   ------------   ----------
  000000          000000          000000         000000

USER      ACCESS   ACCESS COUNT
----      ------   ------ -----
R390      READ        000000
SYSADM    ALTER       000000
SYSOPR    READ        000000
IBMUSER   ALTER       000000
PDXIA     ALTER       000000

   ID     ACCESS  ACCESS COUNT  CLASS                ENTITY  NAME
-------- -------  ------------ -------- ----------------------------------------
NO ENTRIES IN CONDITIONAL ACCESS LIST
READY
```

Thank you for your time!


Rocket®