ibm.com

e-business security for z/OS 1.5

Security Server for z/OS 1.5 Enhancements

Redbooks

International Technical Support Organization

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

ibm.com®
z/OS®
DB2®
IBM®
MVS™
Perform™
RACF®

The following terms are trademarks of other companies:

Intel, Intel Inside (logos), MMX, and Pentium are trademarks of Intel Corporation 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.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

SET, SET Secure Electronic Transaction, and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.

Other company, product, and service names may be trademarks or service marks of others.

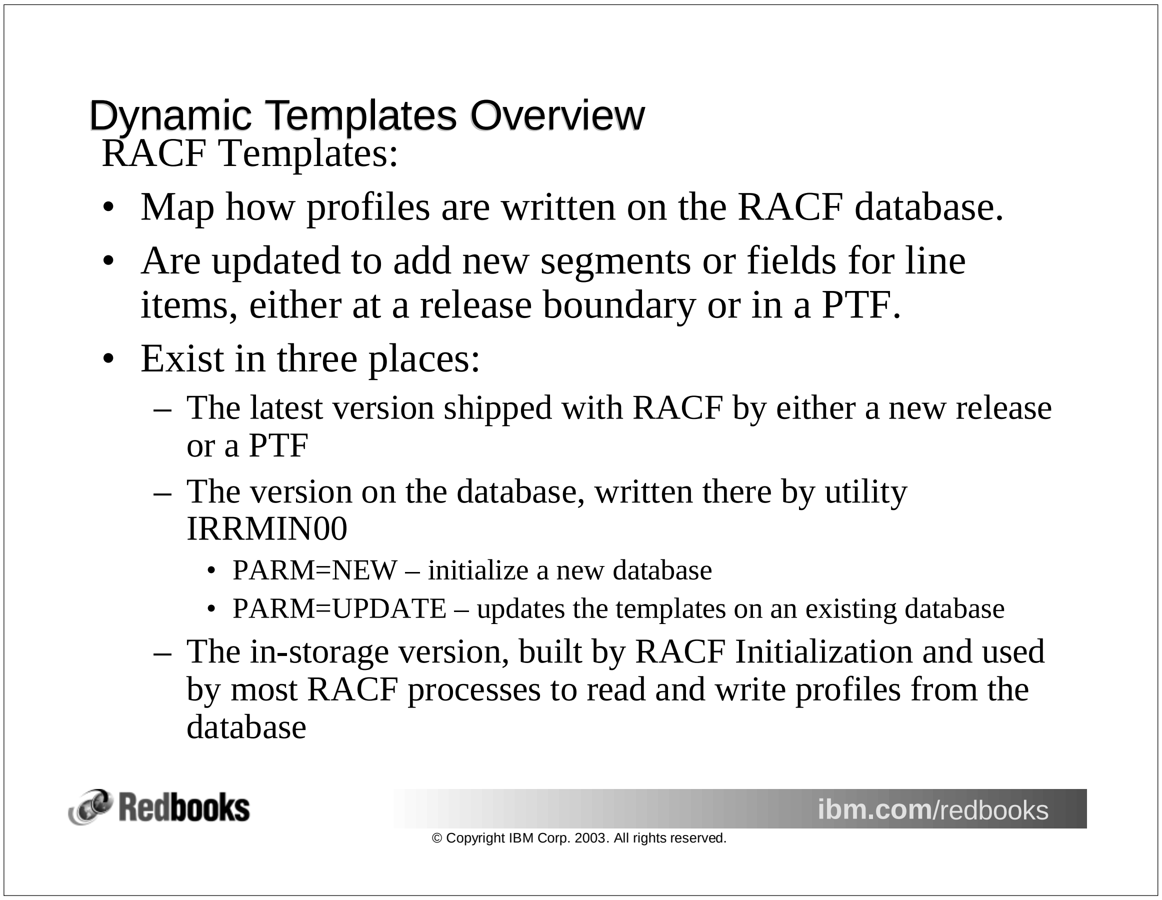
Agenda

RACF Dynamic Templates
RACF/LDAP Event Notification and Password Enveloping
LDAP Change Logging SPE
MLS

IBM®

Security Server RACF Dynamic Templates
(z/OS R5)

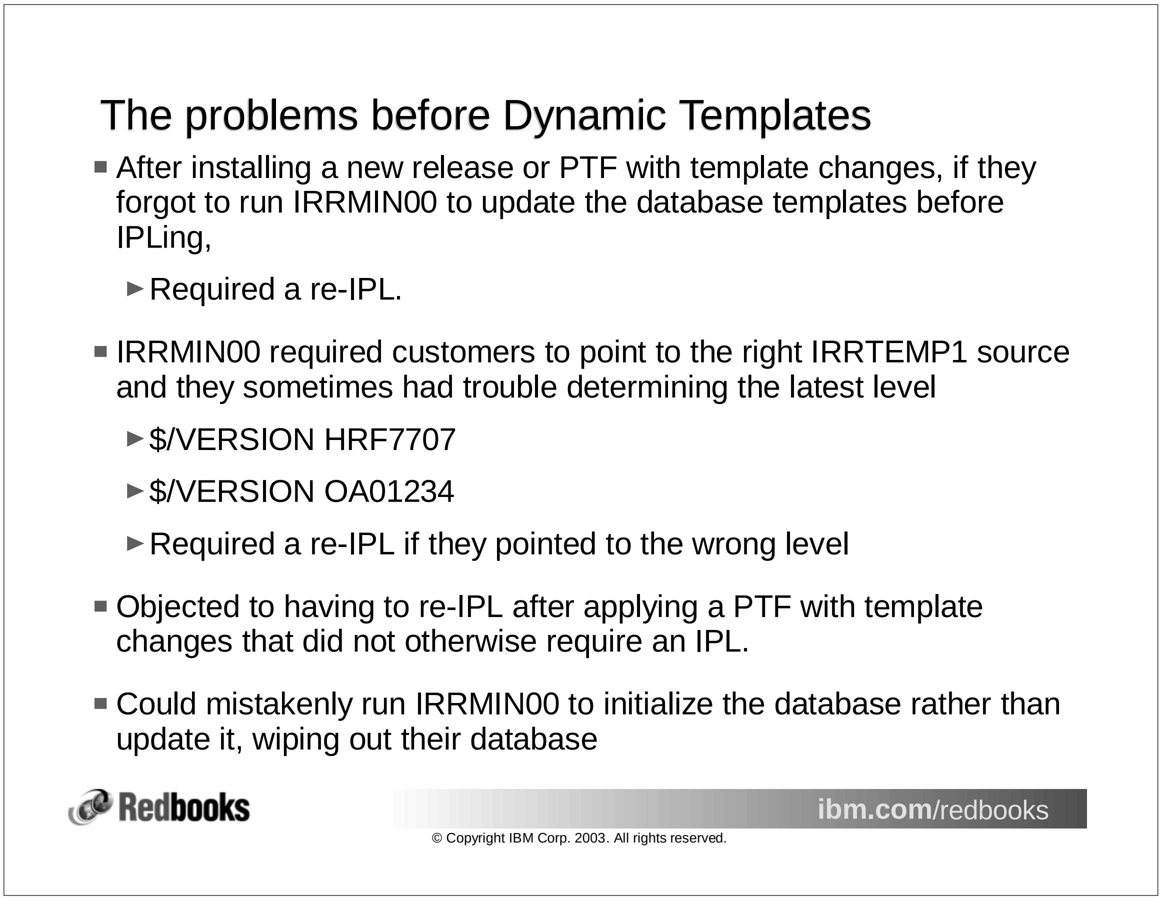
Redbooks
International Technical Support Organization



Dynamic Templates Overview

RACF Templates:

- Map how profiles are written on the RACF database.
- Are updated to add new segments or fields for line items, either at a release boundary or in a PTF.
- Exist in three places:
  - – The latest version shipped with RACF by either a new release or a PTF
  - – The version on the database, written there by utility IRRMIN00
    - PARM=NEW – initialize a new database
    - PARM=UPDATE – updates the templates on an existing database
  - – The in-storage version, built by RACF Initialization and used by most RACF processes to read and write profiles from the database



The problems before Dynamic Templates

- ■ After installing a new release or PTF with template changes, if they forgot to run IRRMIN00 to update the database templates before IPLing,
  - ► Required a re-IPL.
- ■ IRRMIN00 required customers to point to the right IRRTEMP1 source and they sometimes had trouble determining the latest level
  - ► $/VERSION HRF7707
  - ► $/VERSION OA01234
  - ► Required a re-IPL if they pointed to the wrong level
- ■ Objected to having to re-IPL after applying a PTF with template changes that did not otherwise require an IPL.
- ■ Could mistakenly run IRRMIN00 to initialize the database rather than update it, wiping out their database

Dynamic Template Overview

- Have RACF Initialization build the in-storage templates automatically from the latest level whether or not the customer remembered to update the database templates with IRRMIN00 PARM=UPDATE
- Have IRRMIN00 PARM=NEW and PARM=UPDATE automatically write the latest level of templates to the database.
- Do not allow IRRMIN00 PARM=UPDATE to down-level the templates on the database.
- For the PTF-that-does-not-require-an-IPL case, provide a means of dynamically ‘activating’ new templates by replacing the in-storage templates with the new templates.
- Do not allow an existing, active database to be newly initialized (from the system on which the database is active).

The Templates

- The templates shipped with RACF now:
  - Are no longer shipped in source format as IRRTEMP1
  - Are shipped as a module in compiled format as IRRTEMP2. It is linked into two load modules, the RACF Initialization load module and the IRRMIN00 utility load module. The first data lines are:

```
–   DC    CL80'$/VERSION HRF7708 00000010.00000000'
–  DC    CL80'$*'
–  DC    CL80'$/TEMPLATE 001 GROUP VERSION 1'
–  DC    CL80'$/SEGMENT  001 BASE'
–  DC    CL80'GROUP    001 00 00 00000000 00 TEMPLATE FOR A GROUPX
–        PROFILE'
–  DC    CL80'ENTYPE   002 00 00 00000001 01 ENTRY TYPE'
–  DC    CL80'VERSION  003 00 00 00000001 01 TEMPLATE  VERSION NUMX
–            BER'
–  DC    CL80'SUPGROUP 004 00 80 00000008 FF SUPERIOR GROUP'
```

The Templates

- Contain the release and apar level so RACF can determine the latest level of the templates:
  - $/VERSION FMID/APAR# rrrrrrrr.aaaaaaaa
  - $/VERSION HRF7708 00000010.00000000
  - $/VERSION OA01234 00000010.00000010
  - $/VERSION OA01567 00000010.00000020
  - $/VERSION HRFxxxx 00000023.00000020
- In addition to containing standard template information, also contains alias-field related template extension information needed by the RACF manager.
- If an SPE is shipped to more than one previous release, say Rn and Rn-1, the highest level of templates, Rn, is shipped to both.

The SET LIST command

- The @SET LIST operator command displays information on RRSF usage, but also displays the in-storage template level and the dynamic parse level in effect on the system.
- Prior to this line item the output consisted of the FMID or apar number for both:
  - RACF STATUS INFORMATION:
  - TEMPLATE VERSION - HRF7706
  - DYNAMIC PARSE VERSION - HRF7706
- Now it shows the rrrrrrrr.aaaaaaaa values as well:
  - RACF STATUS INFORMATION:
  - TEMPLATE VERSION - HRF7708 00000010.00000000
  - DYNAMIC PARSE VERSION - HRF7708

What happens at RACF Initialization

- During the IPL RACF Initialization automatically builds the in-storage templates from the latest level of the templates:
  - ►1 If the master primary database level is higher or the same as IRRTEMP2, it builds them from the database as it did prior to Dynamic Template support.
    - The level of the templates on the database is kept in the ICB
    - The level of the templates in IRRTEMP2 is on the version card
  - ►2 Otherwise it builds the in-storage templates from IRRTEMP2 itself and issues a template downlevel warning message
    - This scenario will occur if you have forgotten to update the templates on the database prior to IPLing
    - The correct templates are in-storage. The database templates need to be updated so the utilities work properly. No IPL is necessary

IRRMIN00 - database initialization utility

- Will no longer make use of the SYSTEMP data set, which customers typically pointed to SYS1.MODGEN(IRRTEMP1). Now it gets the templates from IRRTEMP2.
- In addition to the templates, will also write the alias-related template extension information to the database
- Will fail PARM=NEW if the output database is active on the system where IRRMIN00 is invoked.
- Will not apply downlevel templates to a database.
- Will make templates active dynamically for the new PARM=ACTIVATE invocation when the templates on the active master primary database are a higher level than the in-storage templates.

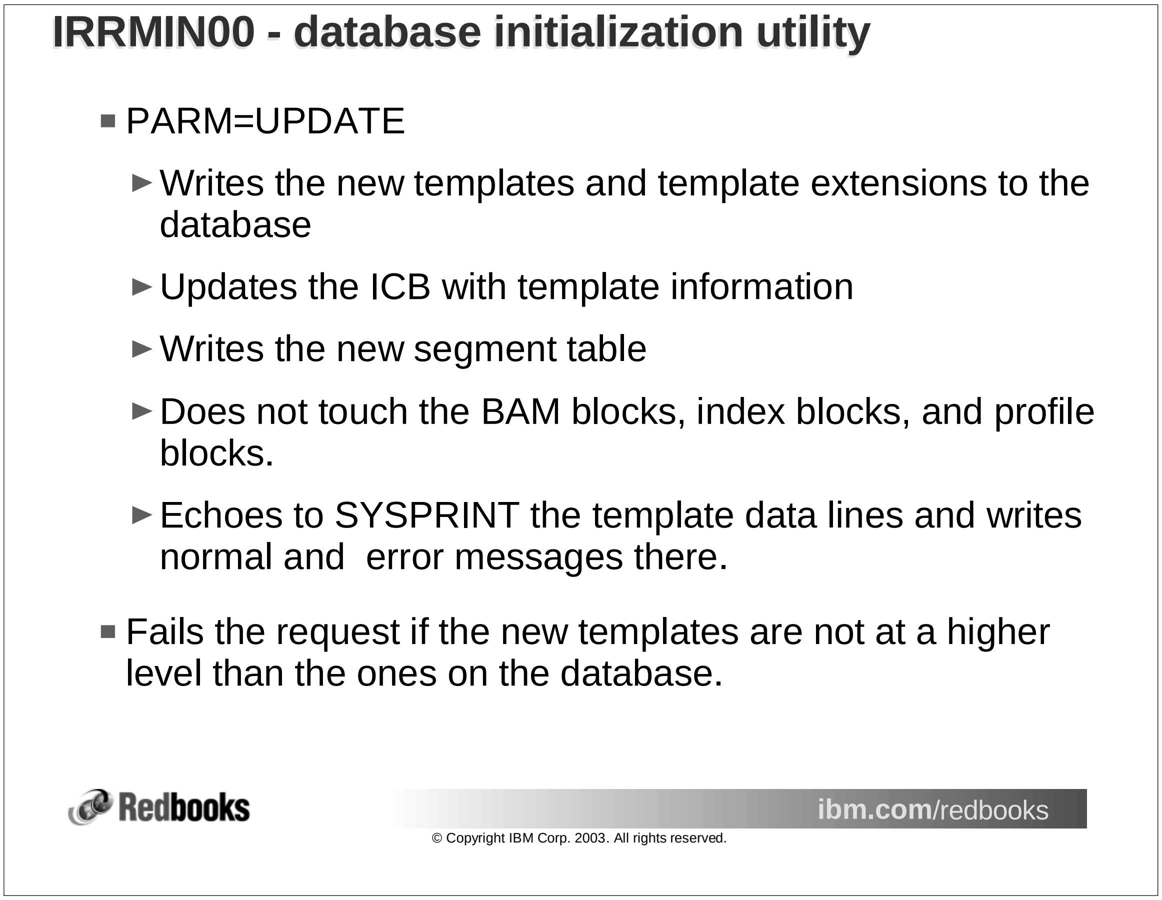
IRRMIN00 - database initialization utility

- PARM=NEW formats a non-VSAM DASD data set as a RACF database. It divides the database into 4K blocks, or records, and initializes them:
  - Block 0: ICB – Inventory control block
  - Block 1 thru Block 10: Templates (and now template extensions)
  - Block 11: Segment table block (becomes the in-storage templates)
  - Block 12 –nn- BAM (block availability mask) blocks
  - Block nn+1 on: Empty blocks for later use as index blocks and profile blocks.
  - It also echoes to the SYSPRINT dataset all the template data lines and writes normal and error messages there.

IRRMIN00 - database initialization utility

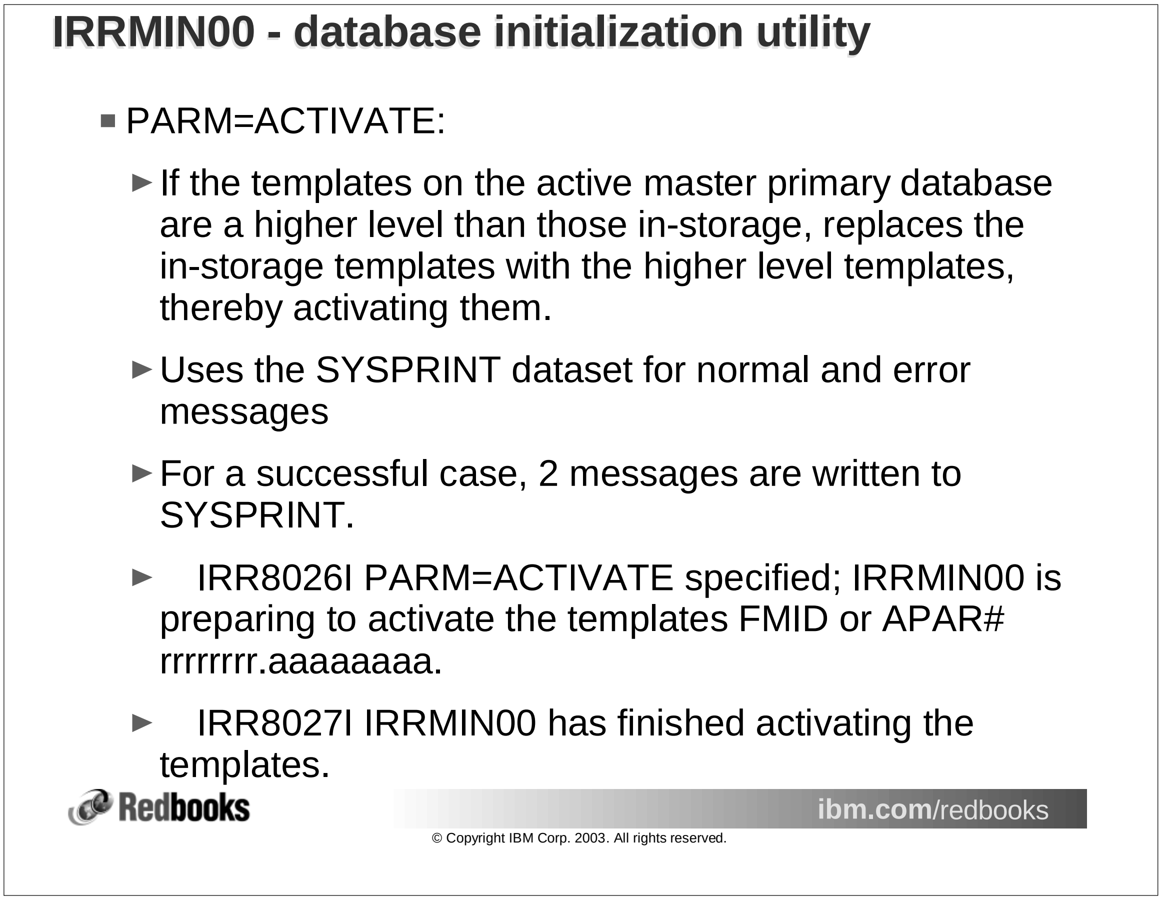
- PARM=NEW will now fail if invoked against an active database on the system where IRRMIN00 is invoked.
  - PARM=NEW sample JCL:

```
//INITRDS  JOB  ,'INITIALIZE NEW DS',
//            MSGLEVEL=(1,1),TYPRUN=HOLD
//INITALZE EXEC PGM=IRRMIN00,PARM=NEW
//STEPLIB  DD  DSN=SYS1.LINKLIB,DISP=SHR,
//            UNIT=YYYY,VOL=SER=YYYYYY
//SYSPRINT DD  SYSOUT=*
//SYSRACF  DD  DSN=SYS1.RACF,DISP=(NEW,CATLG),
//            UNIT=XXXX,VOL=SER=XXXXXX,
//            SPACE=(CYL,(XX),,CONTIG),
//            DCB=DSORG=PSU
```



IRRMIN00 - database initialization utility

- PARM=UPDATE
  - Writes the new templates and template extensions to the database
  - Updates the ICB with template information
  - Writes the new segment table
  - Does not touch the BAM blocks, index blocks, and profile blocks.
  - Echoes to SYSPRINT the template data lines and writes normal and error messages there.
- Fails the request if the new templates are not at a higher level than the ones on the database.



IRRMIN00 - database initialization utility

- PARM=ACTIVATE:
  - If the templates on the active master primary database are a higher level than those in-storage, replaces the in-storage templates with the higher level templates, thereby activating them.
  - Uses the SYSPRINT dataset for normal and error messages
  - For a successful case, 2 messages are written to SYSPRINT.
  - IRR8026I PARM=ACTIVATE specified; IRRMIN00 is preparing to activate the templates FMID or APAR# rrrrrrrr.aaaaaaaa.
  - IRR8027I IRRMIN00 has finished activating the templates.

RRSF Considerations

- If IRRMIN00 PARM=ACTIVATE has been run on one node in an RRSF network, then commands that work on new segments or fields on that system will fail when directed to another node that did not yet have the new templates activated.

Migration/Coexistance considerations

- Continue to run IRRMIN00 with PARM=UPDATE against each of your RACF datasets when upgrading RACF releases or when applying PTFs that update the templates.
  - ► Failure to do so will result in a warning message from RACF at IPL time (ICH579E), and possible error processing from:
    - RACF Database Unload
    - IRRUT200
    - BLKUPD
    - and, some vendor utilites may not function correctly
  - ► UNTIL you run IRRMIN00 PARM=UPDATE
  - ► HOWEVER, normal mainline RACF functions will work properly, and after you run IRRMIN00 PARM=UPDATE, you do not have to re-IPL.
- Later, if installing a PTF with new templates that does not require an IPL, the templates can be activated by running IRRMIN00 PARM=UPDATE followed by IRRMIN00 PARM=ACTIVATE. A re-IPL in order to use the new templates is not necessary.

IBM®

RACF/LDAP Event Notification and Password Enveloping

Redbooks

International Technical Support Organization

LDAP Provides

- Change log support for SDBM(RACF) backend
  - ►Ehanced PC interface for use by RACF
- LDAP interface to retrieve RACF password envelope

RACF Provides

- Creation of LDAP change log entry(via LDAP's enhanced PC interface) when a user profile changes in RACF
- Retrievable user passwords stored in RACF
- R-admin (IRRSEQ00) interface to retrieve encrypted password envelope
- R-Proxyserv (IRRSPY00) interface for applications to create their own change log entries

IDI Provides

- Event handler for polling z/OS LDAP change log
- Java method for decrypting the RACF password envelope
- Sample assembly line which detects a RACF password change, retrieves the password envelope, decrypts it, and applies the password to an entry in IBM Directory Server

LDAP Event Notification

- Enabled by activating new RACFEVNT class and defining NOTIFY.LDAP.USER profile
- Change log entries created for changes to
  - a users password, by any method
  - a users revoke status(FLAG4 field), by any method
  - Other user fields(*) by the ADDUSER, ALTUSER,PASSWORD, and DELUSER commands
- Application changes made using RACROUTE or ICHEINTY not logged
  - Application can call R_Proxyserv to create log entry

LDAP Change log entry contains

- Unique change log entry identifier
- Time and date of change
- Change type(add, modify, delete)
- Change initiator
- Change target
- Does NOT contain details of actual change

Example of change log entry

```
dn: changenumber=13,cn=changelog
objectclass: top
objectclass: changeLogEntry
objectclass: ibm-changelog
changenumber: 13
targetdn: racfid=JOEUSER,profiletype=user,o=ibm,c=us
changetype: modify
changetime: 20030729123000
ibm-changeinitiatorsname:
racfid=JOEADMIN,profiletype=user,o=ibm,c=us
```

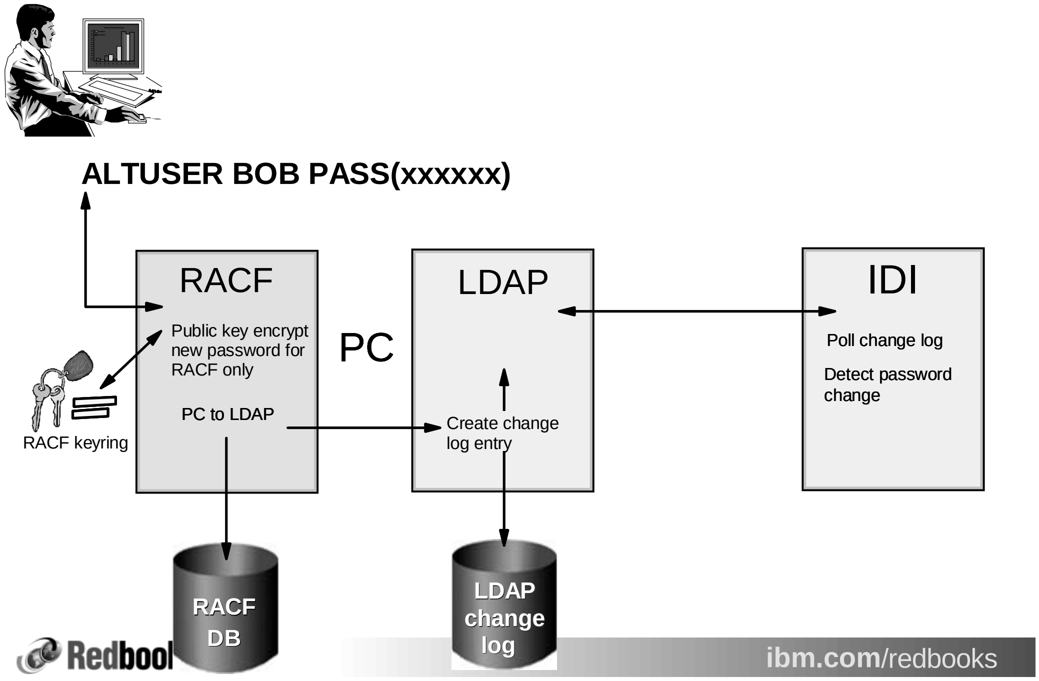
How to setup LDAP notification

- Define NOTIFY.LDAP.USER profile in RACFEVNT class
- Activate RACFEVNT class

Password Enveloping

- New function which allows authorized applications to recover a users clear text password
- A key ring owned by the RACF subsystem contains certificates for password recipients
- LDAP change log entry can be created to log the password update and envelope creation
- Retrieval of envelope controlled by a FACILITY profile

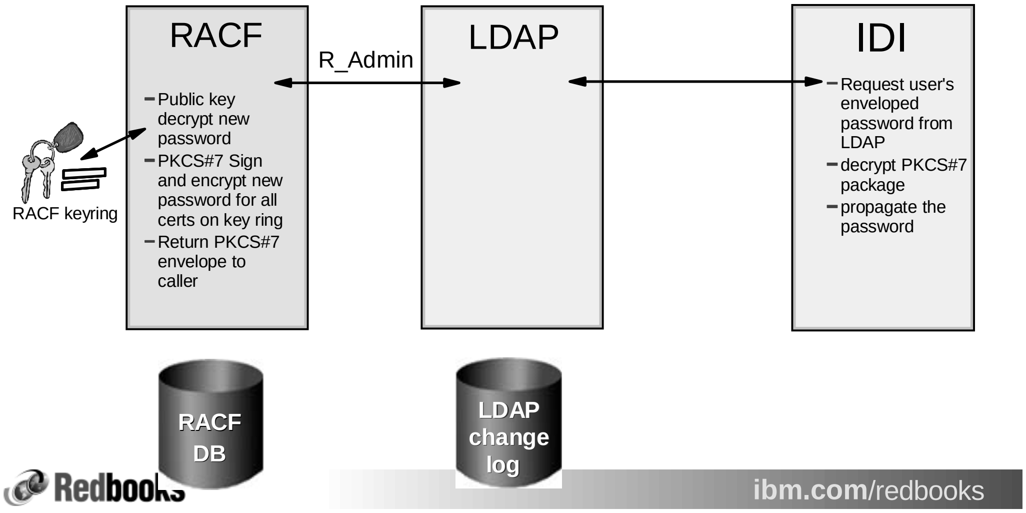
Password update part 1



ALTUSER BOB PASS(xxxxxx)

Password update part 2

ALTUSER BOB PASS(xxxxxx)



RACF
R_Admin
LDAP
IDI
Public key decrypt new password
PKCS#7 Sign and encrypt new password for all certs on key ring
Return PKCS#7 envelope to caller
RACF keyring
Request user's enveloped password from LDAP
decrypt PKCS#7 package
propagate the password
RACF DB
LDAP change log

Contents of the password envelope

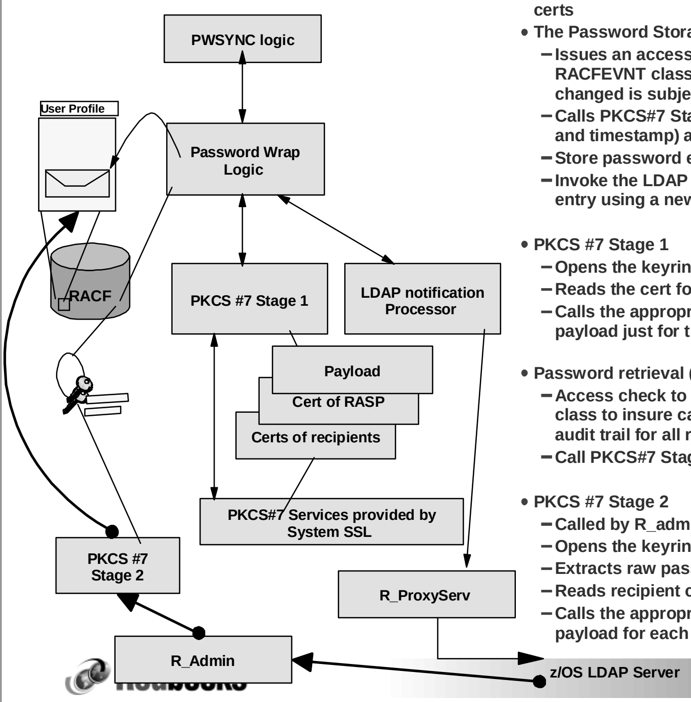
- Password payload is first signed, then encrypted using PKCS#7 functions provided bt System SSL
- payload is BER-encoded ASCII. Password is in lower case. ASN.1 format of the payload is

```
PasswordPayload ::= SEQUENCE{
     Version                    INTEGER
     Expired                    BOOLEAN
     Password                    UTF8String
     Changetime                 IA5String
     Language                   IA5String  OPTIONAL DEFAULT "ENU"
  }
```

PKCS#7 Password Envelope

- PKCS #7 enveloping scheme effectively scopes the processes which can recover a user's password.
  - Each of the processes that are intended recipients of a user's RACF password are identified by a X.509V3 certificate which is populated into a RACF keying
  - This keyring is logically associated with the RACF identity of the RACF remote sharing address space
  - The default key in this keyring is 'owned' by RACF remote sharing address space identity.
- A protected copy of the RACF password is stored in the user's RACF profile
  - The change log serves as an event notification mechanism for the "interested processes"
  - When a password change occurs, RACF:
    - Creates a new copy of the RACF password in the RACF database, secured by the default key in the keyring.
    - Notifies LDAP that a password change occurred.
  - When a change has occurred, the recipient process:
    - Binds to z/OS LDAP
    - Requests a PKCS #7 envelope from RACF
    - The recipient process "opens" the envelop by using the private key associated with the process to successfully decrypt the password related payload.
- The z/OS LDAP server:
  - Upon successful bind, establishes on the thread of execution the RACF security context for the bind specified DN. (existing function)
  - based on the attribute name e.g.racfEnvelopedPassword, will call a new function code of the SAF r_admin interface to retrieve the password blob.
    - At this time, RACF creates a new PKCS#7 envelope which is only readable by the recipients in the keyring. This PKCS#7 envelope is also signed by the default key in the keyring.
    - Binary blob returned by RACF which is returned by LDAP to the requester
- RACF R_admin extension
  - Also levies an access control decision to determine if the Bind DN (as conveyed by the RACF security context on the thread of execution) is authorized to retrieve the racfEnvelopedPassword blob.
    - Auditing of retrieval is performed by r_admin in a consistent fashion as other profile controlled RACF auditing.

Password Envelope Processing



- **The RACF address space has an RACF defined identity-- and this identity is used as a logical anchor for a keyring which contains the recipient certs**
- **The Password Storage/protection Logic:**
  - **Issues an access check to the profile PASSWORD.ENVELOP in the RACFEVNT class to determine if the user whose password is being changed is subject to password enveloping.**
  - **Calls PKCS#7 Stage 1 passing payload (password, password status, and timestamp) and keyring name**
  - **Store password envelope into the USER profile**
  - **Invoke the LDAP notification processor, to create an LDAP change log entry using a new function of the R_Proxyserv callable service**
- **PKCS #7 Stage 1**
  - **Opens the keyring IRR.PWENV.KEYRING**
  - **Reads the cert for the RASP user ID**
  - **Calls the appropriate System SSL PKCS#7 services to envelope the payload just for the RASP.**
- **Password retrieval (via r_admin)**
  - **Access check to IRR,RADMIN.PWENV.EXTRACT profile in FACILITY class to insure caller is allowed to retrieve passwords. Also provides audit trail for all retrieved passwords.**
  - **Call PKCS#7 Stage 2**
- **PKCS #7 Stage 2**
  - **Called by R_admin when recoverable password is requested.**
  - **Opens the keyring IRR.PWENV.KEYRING**
  - **Extracts raw password payload from RACF using RASP private key.**
  - **Reads recipient certs w/o private keys and builds an array of certs**
  - **Calls the appropriate System SSL PKCS#7 services to envelope the payload for each of the recipient, and sign with RASP cert.**

Password enveloping Setup

- Update RACF database templates and IPL
- Implement RACF subsystem address space (RASP)
  - Define OMVS segment for RASP and its groups
  - If RASP not TRUSTED/PRIVILEGED, permit to IRR.DIGCERT.LISTRING in the FACILITY class
- Define Certificate Authority certificate to RACF
- Define Certificate and IRR.PWENV.KEYRING key ring for RASP
- Connect RASP cert to key ring as DEFAULT
- Define/import certificates for recipients and connect them to key ring. Make sure they have TRUSTED status

Password enveloping Setup

- Define PASSWORD.ENVELOPE profile in the RACFEVNT class, activate RACFEVNT
  - Set APPLDATA to configure signature algorithm and encryption strength, or take defaults of MD5 and STRONG
  - Permit/exclude users groups as appropriate. Users with READ access will have new passwords enveloped
  - RACLIST is optional
  - Need to stop and start RASP so UNIX System Services environment can be established
- Permit recipients to IRR.RADMIN.EXTRACT.PWENV in the Facility class

Password enveloping Setup

- If LDAP notification of password changes is required, define NOTIFY.LDAP.USER profile in RACFEVNT
  - ► Change log entry only created for password change if password is enveloped
  - ► Password change log entry contains 'changes' field which indicates that the password has changed, but does not include the actual password or envelope
  - ► If password and non-password are changed in the same command
    - ie ALTUSER CHRIS PASSWORD(NEWPW) RESUME OWNER(VICENTE)
    - two separate change log entries are created

New and changed interfaces

- Update to RACF database templates
  - ‣ Must run IRRMIN00 utility to UPDATE the templates, and re-IPL
  - ‣ Database Unload Utility indicates envelope existence
- New **RACFEVNT** class, and associated profiles
- New **IRR.PWENV.KEYRING** keyring architected for this application
- New R_Admin function to retrieve PKCS#7 password envelope
- New R_Proxyserv (IRRSPY00) function to create LDAP change log entry
- Can be used by applications to log their own changes which 'fly under the radar' of RACF
- Several new warning messages for setup errors and unexpected errors
  - ‣ issued as WTOs
- SET TRACE diagnostic aid

Example of a change log entry for password change

```
dn: changenumber=13,cn=changelog
objectclass: top
objectclass: changeLogEntry
objectclass: ibm-changelog
changenumber: 13
targetdn: racfid=JOEUSER,profiletype=user,o=ibm,c=us
changetype: modify
changetime: 20030729123000
Changes: replace: racfPassword
racfPassword:*ComeAndGetIt*
ibm-changeinitiatorsname: racfid=JOEADMIN,profiletype=user,o=ibm,c=us
```

Not all password changes are enveloped

- For users who do not have READ access to PASSWORD.ENVELOPE
- Initial ADDUSER passwords
- When the new password is the same as the current password
- When the ALTUSER or PASSWORD command is used to change the password, and the new password is equal to the users default grup name
- When an application uses RACROUTE or ICHEINTY to set the password, and the password contains characters which would not be accepted by the RACF commands
- When an application uses RACROUTE or ICHEINTY to set the password and specifies ENCRYPT=NO

IBM®

LDAP Change Logging SPE
(z/OS R5 LDAP)

Redbooks

International Technical Support Organization

LDAP Change logging overview

- RACF can be set up to notify LDAP whenever a RACF user is added, modified, or deleted as discussed earlier
- LDAP creates a change log entry containing the information passed by RACF
  - translated into LDAP format:

```
dn: changenumber=1923,cn=changelog
objectclass: changeLogEntry
objectclass: ibm-changelog
changenumber: 1923
targetdn: racfid=JENSEN,profiletype=USER,cn=myRACF
changetype: modify
changetime: 20021008190331.895766Z
ibm-changeinitiatorsname:
 racfid=ADMIN1,profiletype=USER,cn=myRACF
changes::
replace:racfpasswordenvelope\nracfpasswordenvelope:
  xxxx\n-\n
```

LDAP Change logging overview

- New extended operation to allow an application to request LDAP creates a change log entry
  - ► Currently only used by RACF for changes to the user profile
- Change log entries contain details about changes to data controlled by an application
  - ► Each entry is identified by an increasing change number
  - ► Change log implemented as a new backend, GDBM
    - – Similar to TDBM but GDBM is only for change log entries
  - ► Uses DB2 to store change log entries
  - ► Configuration options for controlling the size of the change log and switching change logging on/off
- Enhanced LDAP searching to retrieve RACF user password envelope

LDAP Change logging Migration

- Rename any existing suffix that overlaps cn=changelog
  - ► Do this before configuring the change log
  - ► modRdn operation can be used
- Additional attributes and objectclasses in TDBM minimum schema
  - ► Existing TDBM schema updated automatically by LDAP server
- Sysplex considerations
  - ► When sharing change log database
    - – Each server should configure change logging the same way
    - – Servers can turn change logging off to avoid logging changes
    - – Change logging must be on in the system where RACF changes are logged

LDAP Change logging startup steps

- Update the LDAP server configuration file
  - ►Add GDBM backend section
    - –Set the change log size limits
    - –Start change logging
  - ►Add SDBM backend section
  - ►Add listen statement to enable PC Callable support
- Create DB2 database for the change log
  - ►Update SPUFI scripts with the unique database owner and name
- Perform RACF configuration for creating change log entries and creating/retrieving password envelopes
- Start the LDAP server
  - ►Look for successful message to say change logging is enabled(GLD0244I)

LDAP Change logging Usage

- Search rootDSE to determine location of the change log plus the range of change log entries
- If the changes attribute within a change log entry contains:
  - ►RACF password was changed
  - ►Use the SDBM search to retrieve RACF envelope containing new password
    - –Specify racfPasswordEnvelope in the attributes to be returned
    - –The returned value is binary base-64 encoded
- If no changes attribute is in the change log entry with changetype=MODIFY:
  - ►Other RACF user values chnaged
  - ►Use SDBM to retrieve user values and process them
- Applications should delete change log entries that are no longer required

LDAP Change logging GDBM configuration

- Add GDBM section in the configuration file
- Enter the required GDBM configuration options:
  - ►database GDBM GLDFDBM[name]
  - ►dbuserid dbowner
  - ►servername
- Other non required GDBM options:
  - ►attrOverflowSize, dsnaoini,include,multiserver,readOnly,sizeLimit, and timeLimit

LDAP Change logging GDBM configuration options

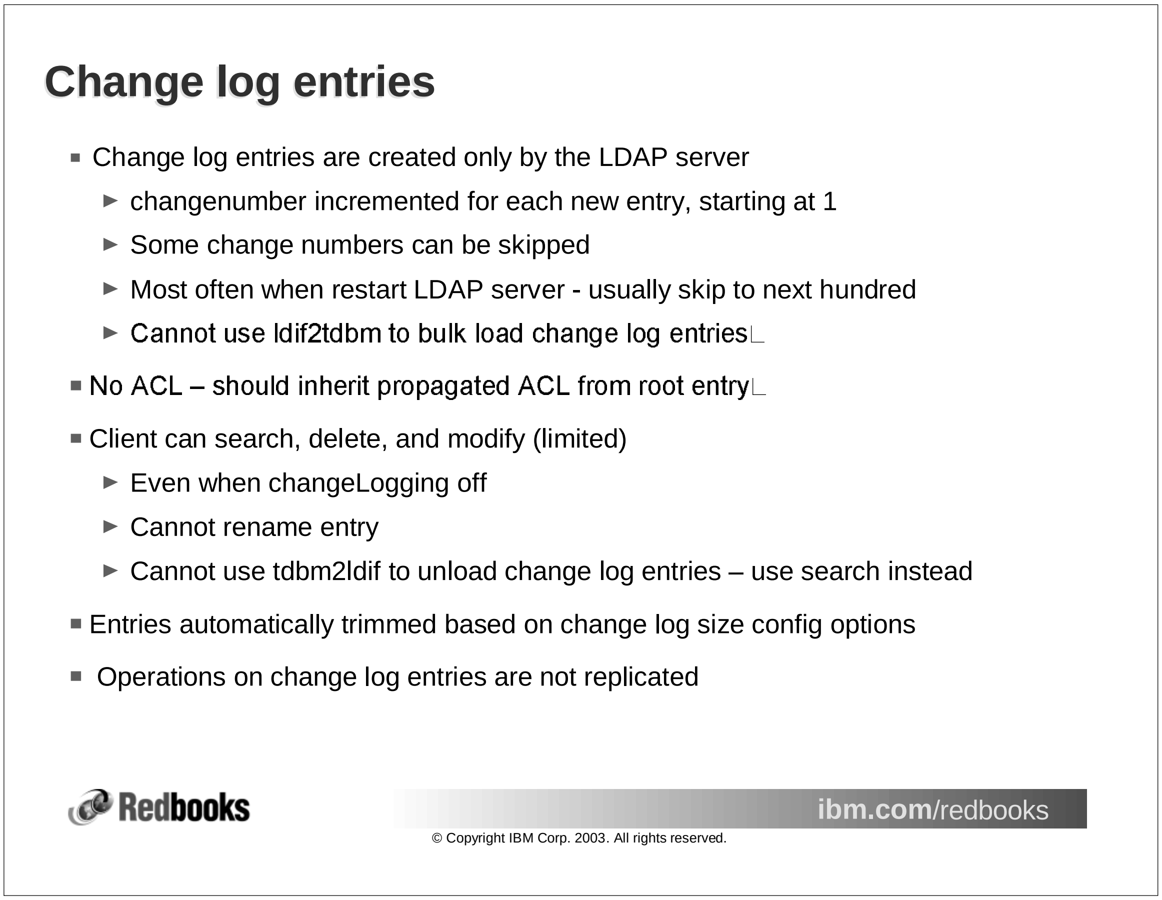
- changeLogging on|off - starts/stops change logging
  - ► When off
    - Client can search/modify/delete change log entries
    - No new entries created and no entries trimmed (deleted) by server
  - ► Default: on
- changeLogMaxAge nnn - maximum age in seconds of a change log entry
  - ► Value can be 0 – 2147483647
  - ► Change log entries older than this are trimmed by the server, unless changeLogging off
  - ► Default: 0 (no maximum)
- changeLogMaxEntries nnn - maximum number of entries in change log
  - ► Value can be 0 – 2147483647
  - ► When reaches this number of entries, lowest numbered entries are trimmed until reduce to 95% of max value, unless changeLogging off
  - ► Default: 0 (no maximum)

LDAP Change logging Additional configuration steps

- SDBM backend section must be included in configuration file
  - ► SDBM suffix used to form RACF DNs when creating change log entry
  - ► SDBM needed to retrieve RACF user password envelope∟
- LDAP Program Callable support must be enabled
  - ► Used by RACF to communicate with LDAP Server
  - ► Specify following option in global section of config file or on server start command:
    - listen ldap://:pc
- Create DB2 database for GDBM
  - ► Create RACF userid to own the database
  - ► Run TDBM SPUFI scripts
    - Update with userid and unique database base
    - Cannot share database with TDBM

Change log entries

- Multiple change log entries
  - ► Each is a leaf directly under the change log root entry
- DN is: changenumber=nnn,cn=changelog
- Each contains information about one RACF user profile change
  - ►dn: CHANGENUMBER=1815,CN=CHANGELOG
  - ►objectclass: CHANGELOGENTRY
  - ►cn: IBM-CHANGELOG
  - ►targetdn: racfid=U12345,profiletype=user,CN=MYRACF
  - ►changetime: 20030611161820.374472Z
  - ►changetype: MODIFY
  - ►changes: replace: racfpassword
  - ►racfpassword: *ComeAndGetIt"
- ibm-changeinitiatorsname: racfid=SUADMIN,profiletype=user,CN=MYRACF
- Changes attribute only present for a password change

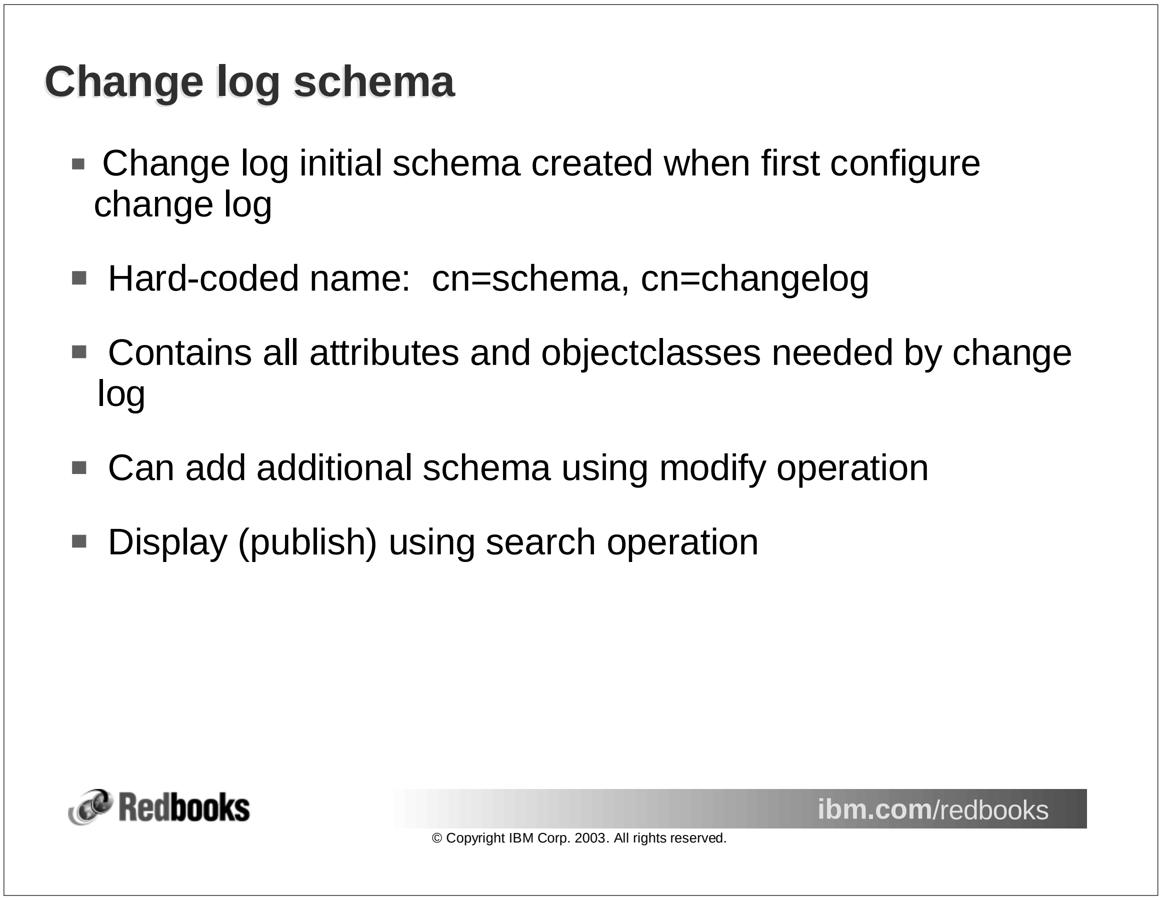


Change log entries

- Change log entries are created only by the LDAP server
  - changenumber incremented for each new entry, starting at 1
  - Some change numbers can be skipped
  - Most often when restart LDAP server - usually skip to next hundred
  - Cannot use ldif2tdbm to bulk load change log entries∟
- No ACL – should inherit propagated ACL from root entry∟
- Client can search, delete, and modify (limited)
  - Even when changeLogging off
  - Cannot rename entry
  - Cannot use tdbm2ldif to unload change log entries – use search instead
- Entries automatically trimmed based on change log size config options
- Operations on change log entries are not replicated

Redbooks

ibm.com/redbooks



Change log schema

- Change log initial schema created when first configure change log
- Hard-coded name: cn=schema, cn=changelog
- Contains all attributes and objectclasses needed by change log
- Can add additional schema using modify operation
- Display (publish) using search operation

Redbooks

ibm.com/redbooks

Change log searches

- Search using standard LDAP search APIs or command utilities
  - ► Use any LDAP client, any platform
- Use any change log attribute in search filter except changes
  - ► Common filter is “changenumber >= nnn”
    - nnn is largest previously retrieved changenumber
- Change log entries returned in increasing changenumber order
- User cannot count on changenumbers not being skipped
- Can search even if changeLogging off

Change log trimming

- Change log entries can be periodically removed by LDAP server
- Based on change log limits set in GDBM section of configuration file
  - ► changeLogMaxAge nnn
    - Server removes entries after in change log more than nnn seconds
  - ► changeLogMaxEntries nnn
    - When surpass nnn entries in change log, server removes entries until reduce to 95% of nnn
    - Lowest numbered entries removed
- Change log checked for trimming when:
  - ► Start server
  - ► New change log entry is created
  - ► Periodically depending on changeLogMaxAge value
- Trimming is not done when changeLogging off

retrieve of RACF user password envelope

- Issue search to the z/OS LDAP Server for the RACF password envelope:

```
>ldapsearch -D racfid=metaAdmn,profiletype=user,cn=myRACF
   -w password  -b racfid=jensen,profiletype=user,cn=myRACF
"objectclass=*" racfpasswordenvelope

racfid=jensen,profiletype=user,cn=myRACF
racfpasswordenvelope::RACF_password_envelope
```

- LDAP contacts RACF for the password envelope
  - Through extensions to the r_admin callable service
- RACF checks requestor authority to see the password envelope
  - LDAP bind DN is mapped to a RACF User ID
- Envelope is only kept in RACF, not in LDAP server
  - Constructed on demand so that the list of reciepents accruately reflects the certificates in the keyring associated with the RACF address space
- Can issue search from any platform

LDAP change logging Application interface

- New LDAP extended operation and response to request LDAP create a change log entry
- changeLogAddEntry – extended operation to create a change log entry
  - Only accepted via LDAP Program Call Interface
  - LDAP Server must be running with change logging configured and started
  - Caller must be in supervisor state
  - Caller provides input to create change log entry attributes
- changeLogAddEntryResponse – returns return code to caller

Obtaining change log status

- Issue search to the z/OS LDAP Server for the root DSE entry:

```
>ldapsearch -D cn=admin  -w password  -V 3 -s base -b ""
"objectclass=*"
```

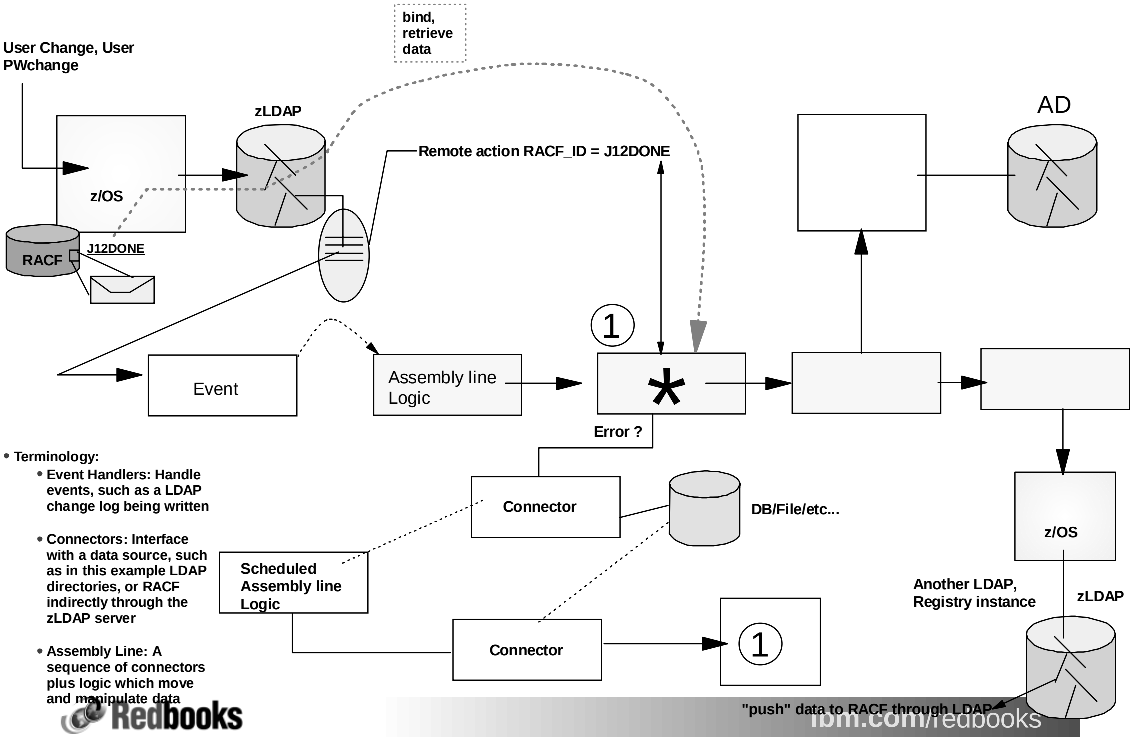
- Results include:

```
changelog=CN=CHANGELOG
firstchangenumber=nnn
lastchangenumber=mmm
```

- Indicates the change log is configured but not if it is started
  - could be **changeLogging off**

Redbooks ibm.com/redbooks

How IBM Directory Integrator (IDI) uses this function...


bind, retrieve data
User Change, User PWchange
zLDAP
AD
Remote action RACF_ID = J12DONE
z/OS
RACF
J12DONE
1
Event
Assembly line Logic
*
Error ?
Connector
DB/File/etc...
z/OS
Scheduled Assembly line Logic
Another LDAP, Registry instance
zLDAP
Connector
1
"push" data to RACF through LDAP

- Terminology:
  - Event Handlers: Handle events, such as a LDAP change log being written
  - Connectors: Interface with a data source, such as in this example LDAP directories, or RACF indirectly through the zLDAP server
  - Assembly Line: A sequence of connectors plus logic which move and manipulate data

Redbooks ibm.com/redbooks

Risks and Rewards

- The notion of password synchronization -- that is synchronizing a user's password across systems and applications needs to be considered carefully from an Enterprise security policy perspective
- The rewards of reduced help desk calls for password resets, and the potential for improved end user customer satisfaction needs to be weighed against the possible risk of password compromise
  - The security characteristics and trust placed in applications and systems which are participating with user account and password synchronization needs to be carefully reviewed and considered
  - A user's password if compromised, is compromised for all systems/applications which that user can access (that are participating in user password synchronization)
    - Understand the potential for loss of business assets accessible by the users
- Recommend at minimum "common sense" approach of higher frequency of password changes -- and longer password histories (on applications/platforms which support policies of these nature)
- If z/OS participates -- passwords limited in length to 8 characters, case insensitive (requirement for longer passwords on z/OS understood)
- This support is granular to the user level -- suggest that highly privileged users are excluded from password synchronization -- however some aspects of account synchronization may indeed be appropriate and of low risk

IBM®

Multilevel Security
(z/OS R5)

Redbooks

International Technical Support Organization

What is Multilevel Security

- Multilevel Security is a security policy that allows the classification of data and users based on a system of hierarchical security levels combined with a system of non-hierarchical security categories.
- Characteristics of a multilevel-secure system:
  - Access controls
    - Mandatory Access Control (MAC)
    - Discretionary Access Control (DAC)
  - Accountability
    - Auditing
    - Identification and Authentication
  - Trusted Computing Base
    - Hardware
    - Software

Why Multilevel Security

- Multilevel Security provides a way to segregate users and their data from other users and their data regardless of access lists, UACC, etc.
- Valuable to government agencies
  - Use of functions like name-hiding, write-down, *-property (no write-down)
- Valuable to commercial customers (i.e. service bureau)
  - Can be set up using a small set of SECLABELs and few SETROPTS options (MLACTIVE and SECLABELCONTROL).
    - Example: MVS system with HTTP Server
    - Assign a “low” SECLABEL to external customers so they can access “external” data
    - Assign a “high” SECLABEL to employees so they can access both “internal” and “external” data

History and Evolution of Multilevel Security

- 1990
  - MVS/ESA 3.1.3 with RACF 1.9, TSO/E, JES, DFP, VTAM, PSF etc., passes formal B1 evaluation having met the criteria specified in the Department of Defense Trusted Computer System Evaluation Criteria, DoD 5200.28-STD.
- 1996
  - Some new functions added to OS/390, such as UNIX, Extended Consoles, TCP/IP, “not designed for B1”.
- 2004
  - With z/OS V1R5 “B1” support is extended to cover these functions.

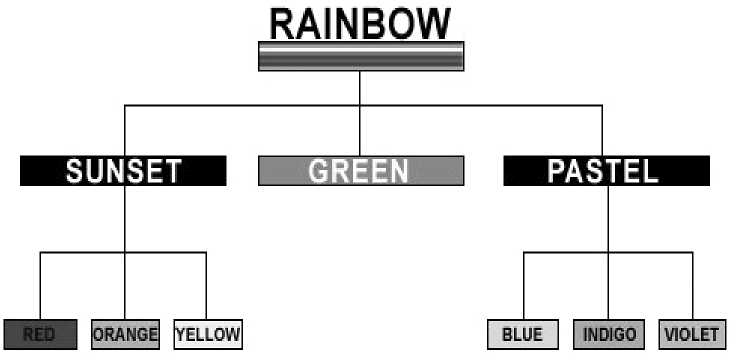
Existing B1 support (before z/OS V1R5)

- RACF and other evaluated system components support Security Labels (a.k.a. SECLABELs).
- SECLABELs have two components:
  - Level (a number in the range 1-255)
    - Unclassified/1
    - Sensitive/25
    - Confidential/50
    - Secret/100
  - List of Categories (0 or more named categories)
    - Green
    - Yellow, Orange
    - Yellow, Orange, Red

Existing B1 support

- Special system-defined SECLABELs
  - SYSNONE
    - Combines the lowest Security Level and has NO Categories
  - SYSLOW
    - Combines the lowest Security Level and has NO Categories
  - SYSHIGH
    - Combines the highest Security Level and ALL Categories

SECLABEL Hierarchy

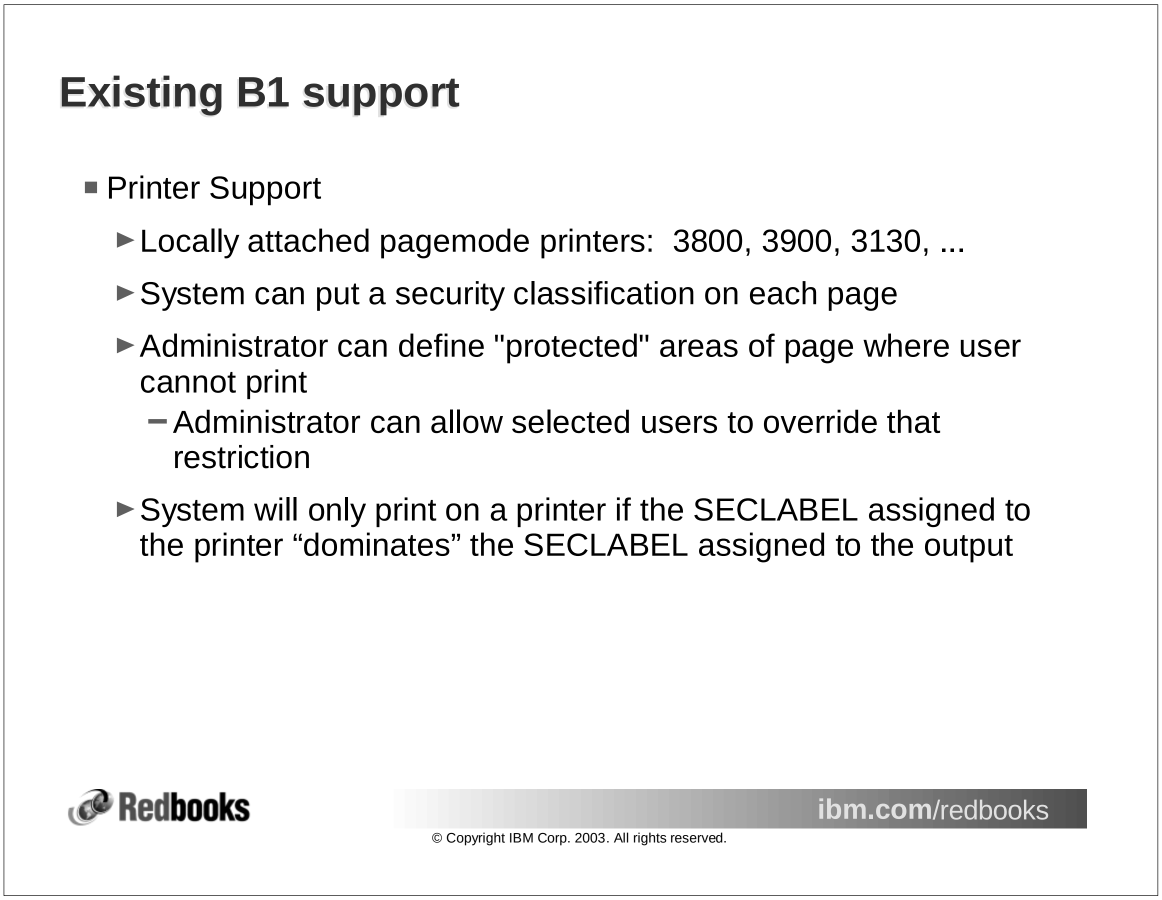


Existing B1 support

- Each user has a default SECLABEL
- Some applications (TSO/E, batch jobs) support user requesting a specific SECLABEL
- Each port of entry (TERMINAL, card reader, ...) has a SECLABEL
- Each SECLABEL has a RACF profile
  - Access list
  - Universal access
  - Auditing information
- During user authentication, RACF validates user's requested SECLABEL
  - User must have access to that SECLABEL
  - SECLABEL must properly match the port of entry

Existing B1 support

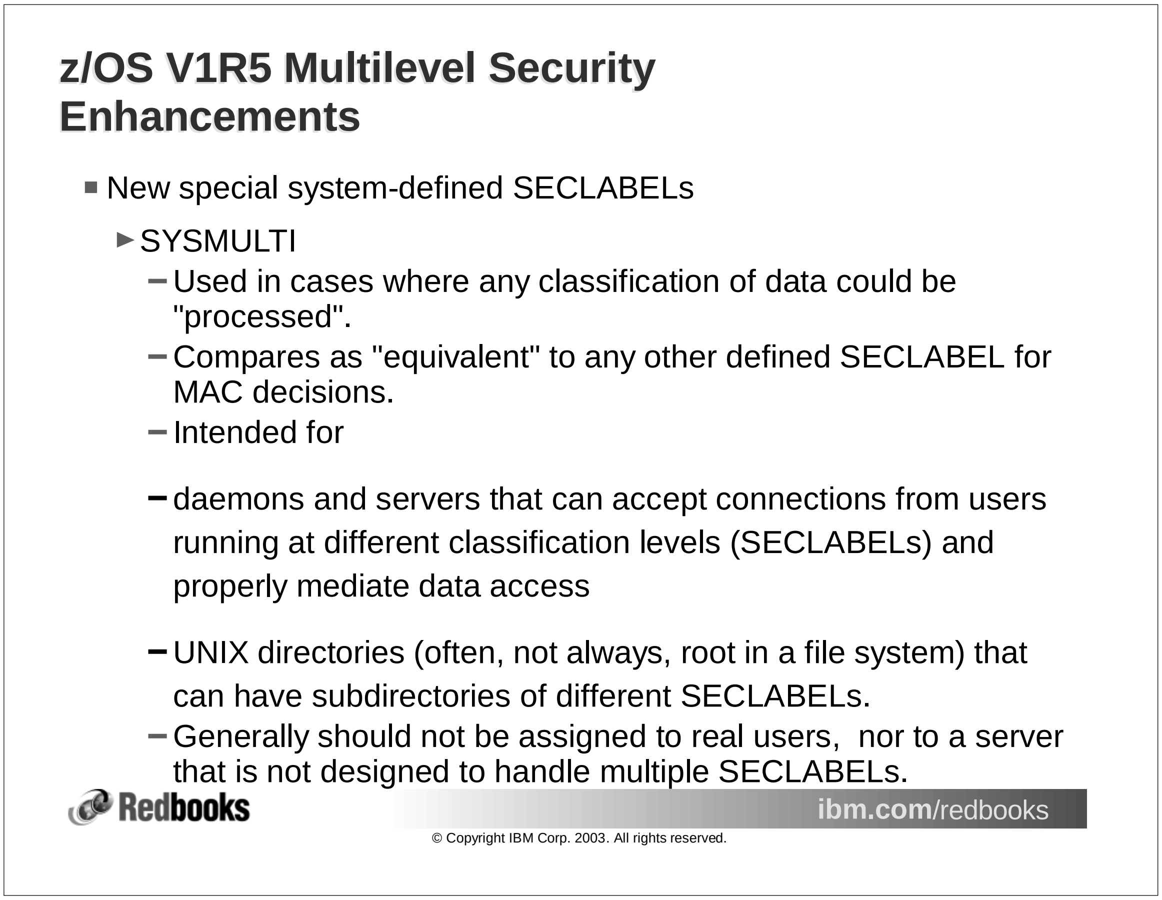
- Various options to control such functions as
  - Whether users and resources must have SECLABELs or not
  - Whether write-down is allowed or not (system wide option)
  - How auditing should be performed
- Authorization checking:
  - User tries to access a resource
  - RACF compares resource SECLABEL with user's SECLABEL (MAC)
  - If that passes, RACF checks access list and universal access (DAC)
  - If that passes, RACF grants access



Existing B1 support

- Printer Support
  - ►Locally attached pagemode printers: 3800, 3900, 3130, ...
  - ►System can put a security classification on each page
  - ►Administrator can define "protected" areas of page where user cannot print
    - −Administrator can allow selected users to override that restriction
  - ►System will only print on a printer if the SECLABEL assigned to the printer “dominates” the SECLABEL assigned to the output

Redbooks ibm.com/redbooks



z/OS V1R5 Multilevel Security Enhancements

- New special system-defined SECLABELs
  - ►SYSMULTI
    - −Used in cases where any classification of data could be "processed".
    - −Compares as "equivalent" to any other defined SECLABEL for MAC decisions.
    - −Intended for
    - −daemons and servers that can accept connections from users running at different classification levels (SECLABELs) and properly mediate data access
    - −UNIX directories (often, not always, root in a file system) that can have subdirectories of different SECLABELs.
    - −Generally should not be assigned to real users, nor to a server that is not designed to handle multiple SECLABELs.

Redbooks ibm.com/redbooks

SECLABELs and MAC checking

- Three types of MAC checking
  - MAC
    - User’s current SECLABEL dominates Resource’s SECLABEL
  - RVRSMAC (Reverse MAC)
    - Resource’s SECLABEL dominates User’s current SECLABEL
  - EQUALMAC (Equal MAC)
    - User’s current SECLABEL is equivalent to the Resource’s SECLABEL.
  - New operand EQUALMAC= added on the ICHERCDE macro
    - EQUALMAC=YES
    - The class requires SECLABEL equivalence

SECLABELs for z/OS UNIX processes and sockets

- Currently TSO/E users:
  - Have the ability to select their current SECLABEL by specifying it on the logon panel, or they can use their default.
  - The value they enter is saved in the TSO segment and used as the default the next time they log on.
- This function has been modified to:
  - Handle workstations (allowing for both reading and writing)
  - Support the z/OS UNIX environment where a user may enter the system from a remote IP address using an application such as rlogin.
  - Associate SECLABELs to IP addresses.

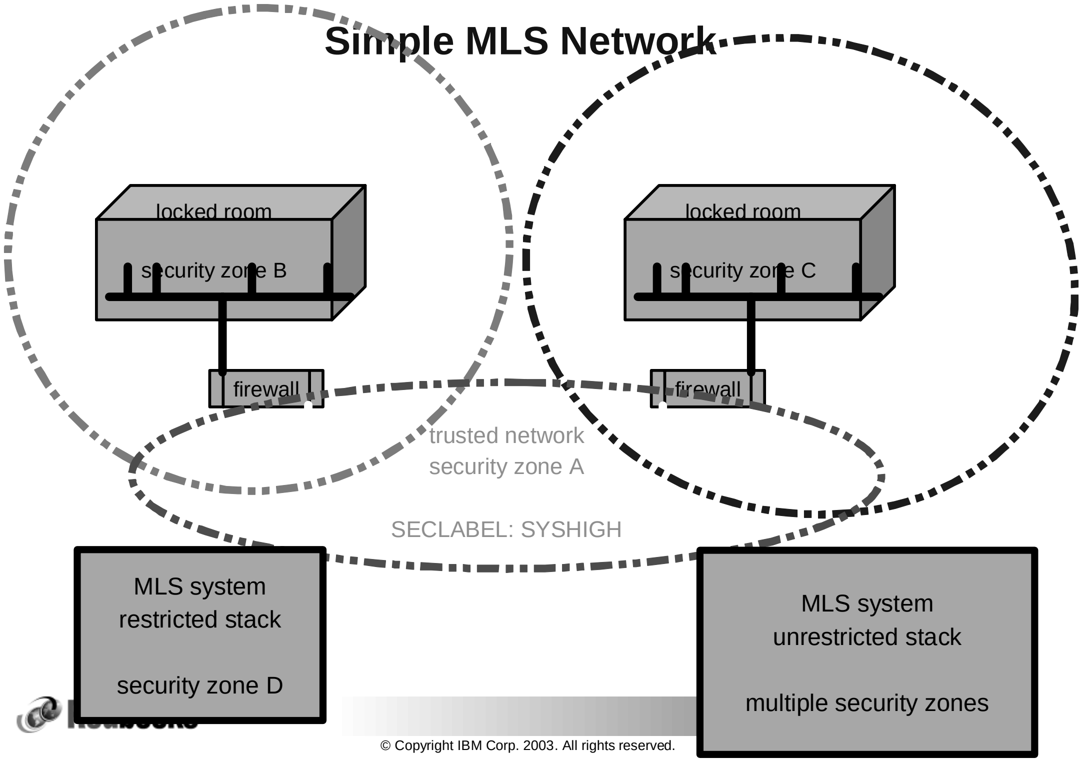
SECLABELs for z/OS UNIX processes and sockets

- SERVAUTH class usage and characteristics have been enhanced to accommodate IP V6 addresses.
- New parameters have been added to InitACEE to allow the SECLABEL and SERVAUTH values to be passed.
- Corresponding changes have been made to allow applications to pass these values through UNIX System Services to InitACEE. These changes accommodate applications willing to change their code to allow the specification of a SECLABEL by the user.
  - New z/OS UNIX callable service, _poe, to set the port of entry for use by servers. Can set TERMINAL or SERVAUTH
  - z/OS UNIX Kernel will provide the server SECLABEL on the User Authentication call

SECLABELs for z/OS UNIX processes and sockets

- Administrator can define IP subnetworks via RACF profiles
  - SERVAUTH class
  - Any granularity desired, down to individual IP address if needed
- SERVAUTH profile contains SECLABEL for that subnetwork
  - Customer responsible for network topology and protection of network links
    - IPSEC (VPN) can also be used to help this
- TCP/IP stack ensures that application on host can only send/receive packets if application and IP address have equivalent SECLABEL
  - Support for servers or daemons that understand MLS (FTP, TELNET, INET)
    - Assign SYSMULTI SECLABEL to server/daemon
    - Can then communicate with any of the subnetworks

Simple MLS Network



locked room

security zone B

firewall

locked room

security zone C

firewall

trusted network
security zone A

SECLABEL: SYSHIGH

MLS system
restricted stack

security zone D

MLS system
unrestricted stack

multiple security zones

SECLABELs for z/OS UNIX processes and sockets

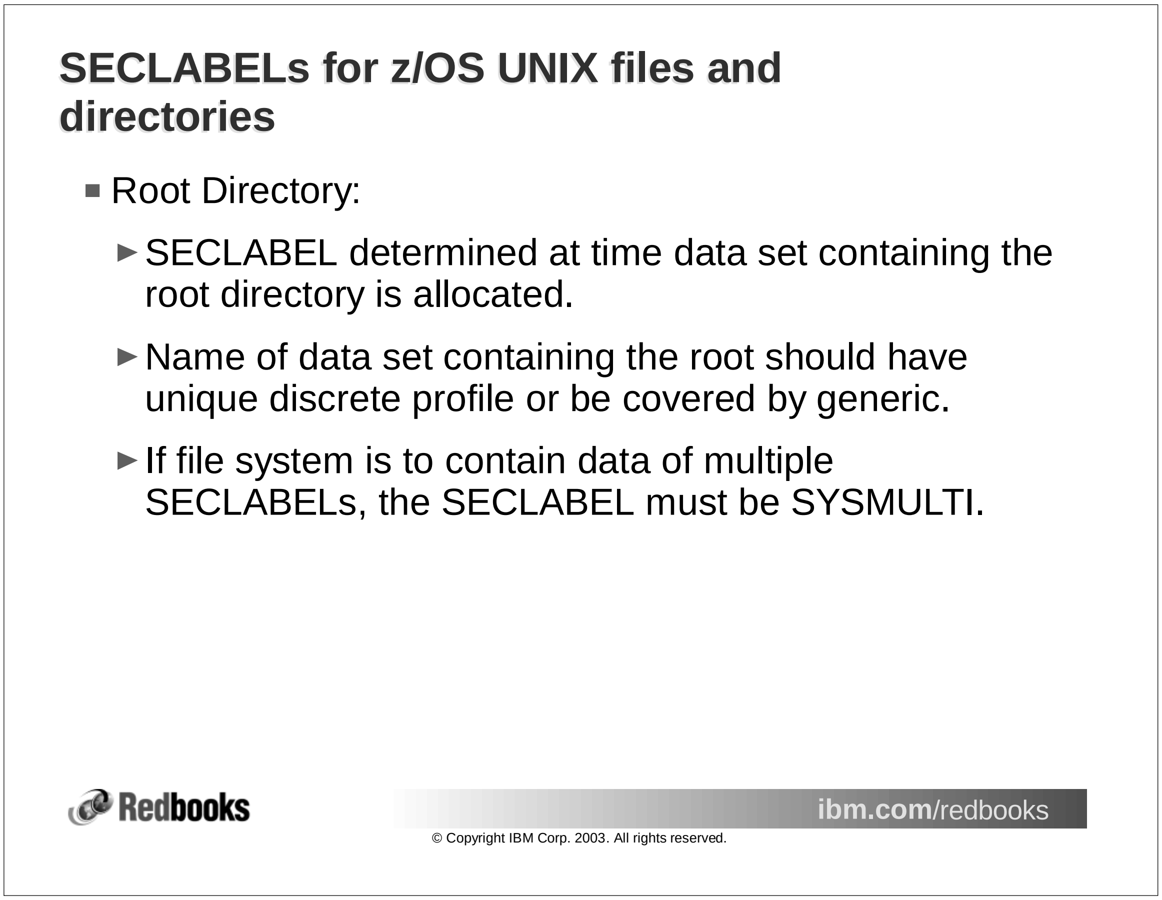
- Program access to SERVAUTH (enhancements to WHEN(PROGRAM) Conditional Access to the SERVAUTH class)
  - Allow appropriate use of PING and TRACEROUTE by a network administrator when multilevel security is enabled
    - Communications Server (TCP/IP) has the ability to restrict access to SERVAUTH resources to users running certain programs
- Allowed ONLY in a “clean environment” (like PADS – Program Access to Data Sets)
  - All programs previously loaded must be program-controlled
  - Uncontrolled programs cannot be loaded into the environment after access has been granted to the SERVAUTH based on the program name

SECLABELs for z/OS UNIX processes and sockets

- Support for SECLABELs for UNIX Processes and Sockets consists of modifications in the following:
  - RACROUTE
  - RACF Manager
  - RACLIST
  - FASTAUTH
  - Callable Service InitACEE
  - Callable Service R_Fork
  - RACF Command Processors:
    - SETROPTS
    - PERMIT
    - RDEFINE, RALTER
  - RACF Utilities UT200, and DB UNLOAD

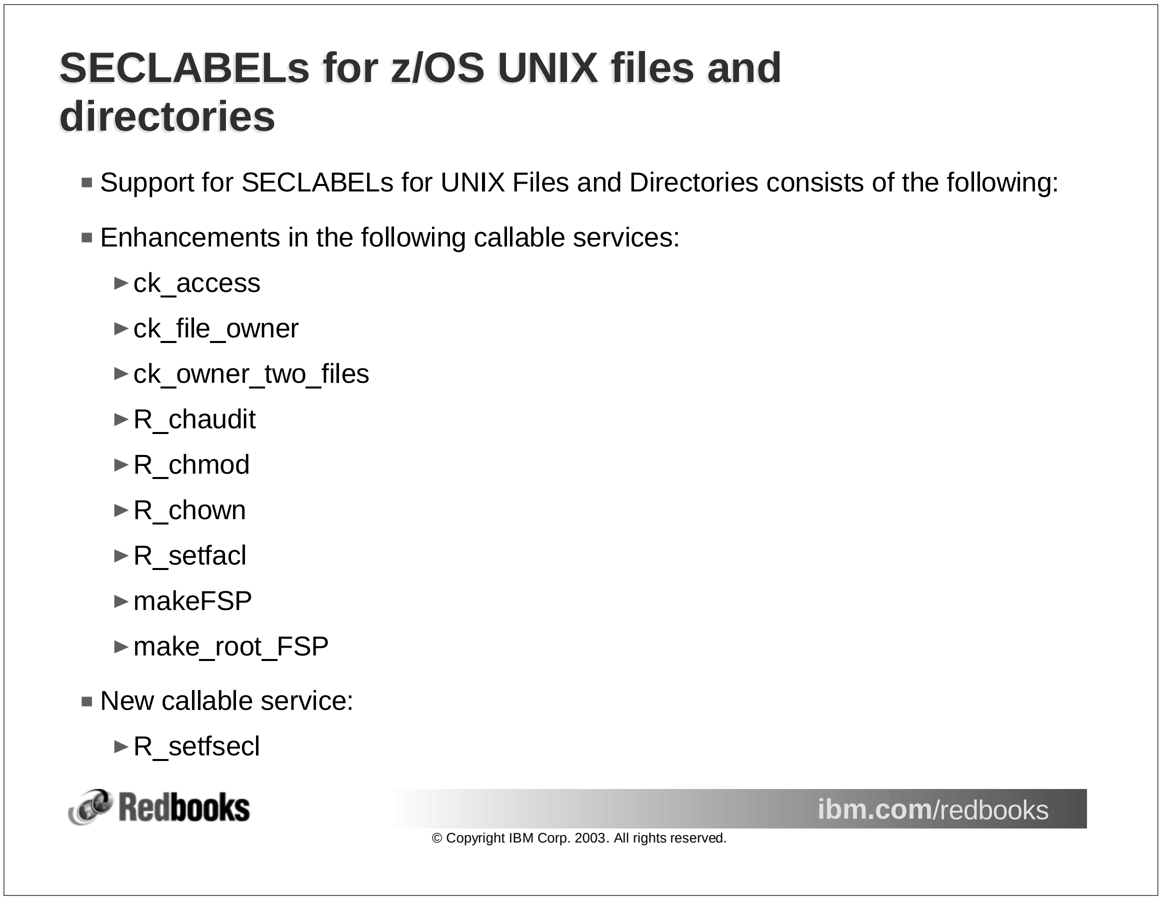
SECLABELs for z/OS UNIX files and directories

- MAC protection for files and directories.
- RACF assigns user's SECLABEL to new file or directory when it is created.
  - SECLABEL cannot be changed.
    - Use the z/OS UNIX command, chlabel, to set one.
    - Copy the file to a directory with the appropriate SECLABEL to change it (subject to dominance and write-down).
  - Subdirectory has same SECLABEL as parent directory (except SYSMULTI).
  - Files in directory have same SECLABEL as directory.
- Enabled/Disabled via the new SETROPTS option MLFSOBJ(ACTIVE / INACTIVE)
  - Requires that UNIX Files and Directories have SECLABELs. It is similar to the existing option MLACTIVE.



SECLABELs for z/OS UNIX files and directories

- Root Directory:
  - SECLABEL determined at time data set containing the root directory is allocated.
  - Name of data set containing the root should have unique discrete profile or be covered by generic.
  - If file system is to contain data of multiple SECLABELs, the SECLABEL must be SYSMULTI.



SECLABELs for z/OS UNIX files and directories

- Support for SECLABELs for UNIX Files and Directories consists of the following:
- Enhancements in the following callable services:
  - ck_access
  - ck_file_owner
  - ck_owner_two_files
  - R_chaudit
  - R_chmod
  - R_chown
  - R_setfacl
  - makeFSP
  - make_root_FSP
- New callable service:
  - R_setfsecl

SECLABELs for z/OS UNIX files and directories

- R_setfsecl
  - New callable service
    - Used to allow for an FSP to be created in an address space other than the user’s.
    - Used to change the SECLABEL of files or directories.
    - Requires
    - A system CRED or,
    - A user CRED for a user with the SPECIAL attribute if no seclabel currently exists in the FSP.
    - To be used ONLY by the Physical File System (z/FS) or z/OS UNIX System Services.
    - Runs in cross-memory
    - Callers must be in Supervisor State

SECLABELs for z/OS UNIX Interprocess Communications

- MAC protection for
  - Pipes
  - UNIX Sockets
- Communication can only occur between processes with equivalent SECLABELs (a.k.a. EQUALMAC).
  - With limited exceptions:
    - The resource or the accessor SECLABEL is SYSMULTI.
- SECLABEL cannot be changed later.
- Enabled/Disabled via the new SETROPTS option MLIPCOBJ(ACTIVE / INACTIVE)
  - Requires that UNIX Interprocess Communications functions (shared memory, message queues, semaphores) have SECLABELs. It is similar to the existing option MLACTIVE

SECLABELs for z/OS UNIX Interprocess Communications

- Support for SECLABELs for UNIX Interprocess Communications consists of :
- Enhancements in the following callable services:
  - makeISP
  - ck_IPC_access
  - ck_IPC_ctl

SECLABEL by System

- Allows customer to share a RACF database between systems and isolate use of specified SECLABELs to specified systems
- Specified by a member list on a SECLABEL profile
  - No members listed
    - Usable anywhere
  - Members listed
    - Usable only on one of those systems
- Not applicable to the seclabels provided by RACF, e.g.
  - SYSHIGH, SYSLOW, SYSNONE, SYSMULTI
- Enabled/Disabled via the new SETROPTS option SECLBYSYS/NOSECLBYSYS

SECLABEL by System

- Example: SECLABELs A, B, and C Systems SYS1 and SYS2
  - Administrator could define them as follows:
    - RDEF SECLABEL (A,B) … ADDMEM(SYS1)
    - RDEF SECLABEL C … ADDMEM(SYS2)
  - Then
    - Any attempt to access system SYS1 using SECLABEL C, or any attempt from SYS1 to access resources with SECLABEL C would fail
    - Any attempt to access system SYS2 using SECLABEL A or B, or any attempt from SYS2 to access resources with SECLABEL A or B, would fail.

SECLABEL by System

- Support for SECLABEL By System entails the following:
  - Enhancements to existing RACF Services:
    - RACROUTE REQUEST=LIST
    - RACROUTE REQUST=VERIFY/VERIFYX
    - RACROUTE REQUEST=EXTRACT
  - Enhancements to existing RACF Commands:
    - RDEFINE
    - RALTER
    - SETROPTS
    - RLIST
    - LISTDSD
    - SEARCH

SECLABEL by System

- New operand SIGNAL= added on the ICHERCDE macro
- Enhancements to SETROPTS processing for SECLABEL By System:
  - New ENF Signal is sent to listeners for those CDT classes that have SIGNAL=YES for
    - SETR RACLIST
    - SETR RACLIST REFRESH
    - SETR NORACLIST
- For the SECLABEL class, allows JES to keep a current list of active SECLABELs by listening for this signal.

Write down by User Privilege

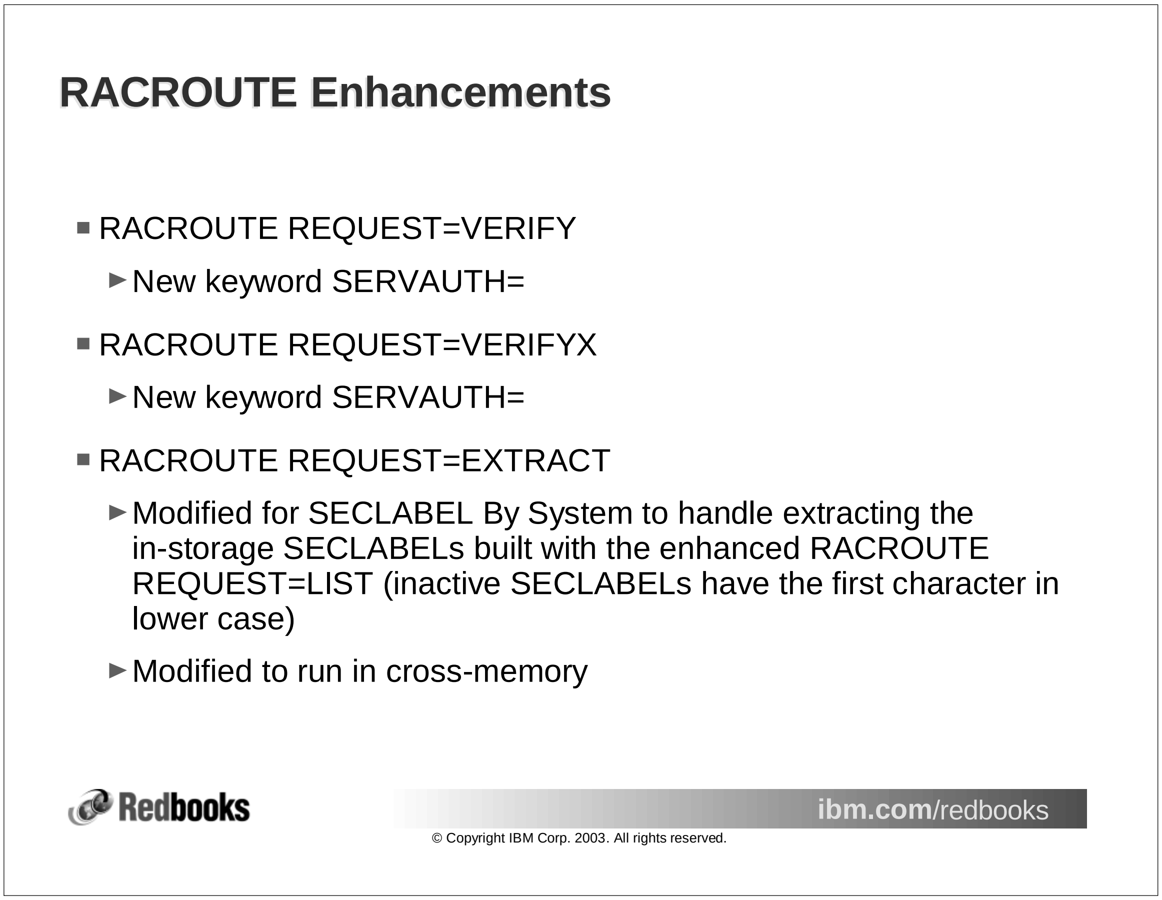
- Allows the Security Administrator to authorize specific users to Write-Down when SETR MLS is in effect.
- R_writepriv
  - New callable service to allow users to dynamically enable, disable, and reset Write-Down.
- RACPRIV
  - New RACF command to provide TSO/E users an interface to the callable service.
- IRR.WRITEDOWN.BYUSER
  - New RACF profile in the FACILITY class, used in the administration of the Write-Down privilege.
- writedown
  - New command for z/OS UNIX users.

Write down by User Privilege

- RACPRIV syntax
  - RACPRIV [WRITEDOWN(ACTIVE| INACTIVE | RESET)]
- RACPRIV WRITEDOWN(ACTIVE)
  - Enables the Write-Down privilege
- RACPRIV WRITEDOWN(INACTIVE)
  - Disables the Write-Down privilege
- RACPRIV WRITEDOWN(RESET)
  - Resets the Write-Down privilege to the default setting
- RACPRIV WRITEDOWN or RACPRIV without any keywords
  - Allows the user to query the current setting

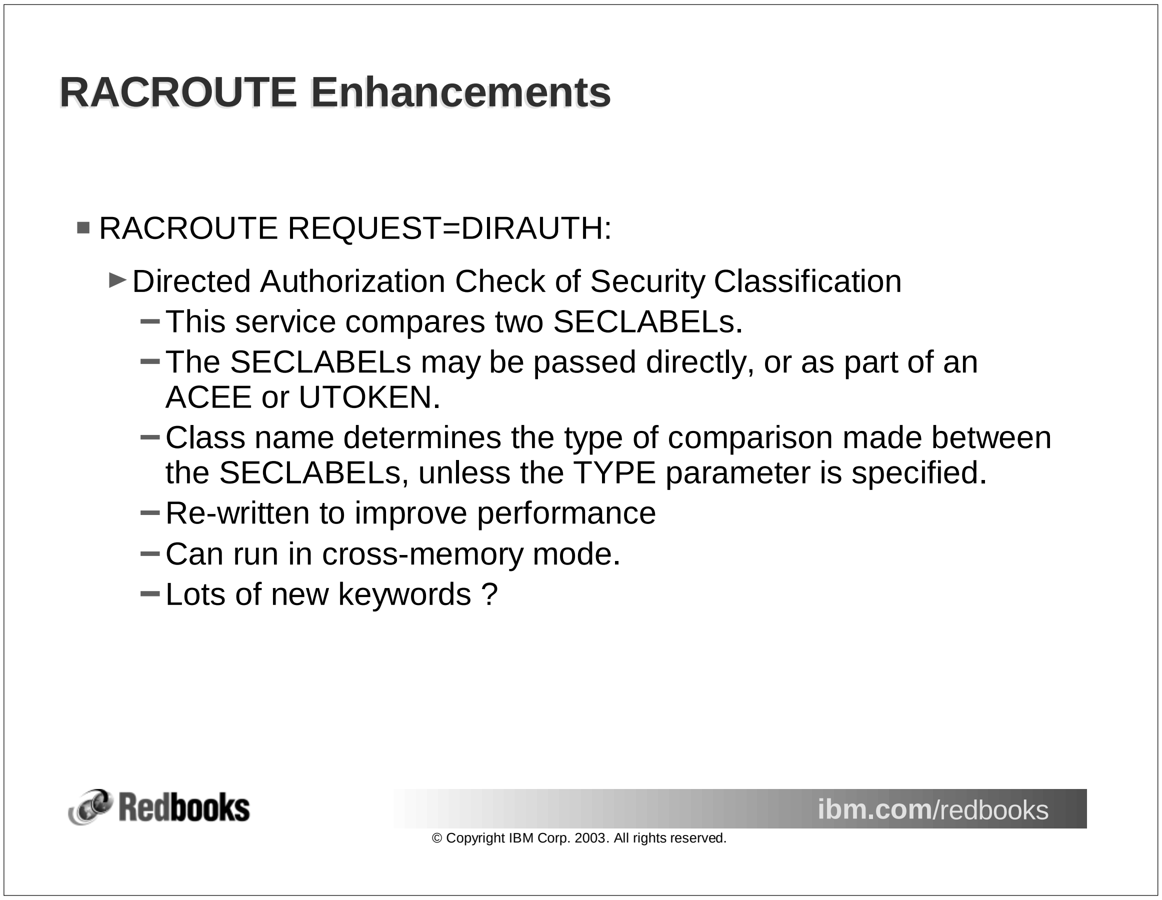
Name Hiding

- Allows installations to prevent users from discovering data set names, file names, and directory names that they didn't already know.
- Enabled/Disabled via the new SETROPTS option MLNAMES/NOMLNAMES
- Needed only if
  - The dataset names contain sensitive data
  - The file names contain sensitive data
- Should not be enabled, unless necessary, because it can cause performance degradation



RACROUTE Enhancements

- RACROUTE REQUEST=VERIFY
  - New keyword SERVAUTH=
- RACROUTE REQUEST=VERIFYX
  - New keyword SERVAUTH=
- RACROUTE REQUEST=EXTRACT
  - Modified for SECLABEL By System to handle extracting the in-storage SECLABELs built with the enhanced RACROUTE REQUEST=LIST (inactive SECLABELs have the first character in lower case)
  - Modified to run in cross-memory



RACROUTE Enhancements

- RACROUTE REQUEST=DIRAUTH:
  - Directed Authorization Check of Security Classification
    - This service compares two SECLABELs.
    - The SECLABELs may be passed directly, or as part of an ACEE or UTOKEN.
    - Class name determines the type of comparison made between the SECLABELs, unless the TYPE parameter is specified.
    - Re-written to improve performance
    - Can run in cross-memory mode.
    - Lots of new keywords ?

RACROUTE Enhancements

- RACROUTE REQUEST=DIRAUTH
  - New keywords
    - TYPE=MAC, EQUALMAC, RVRSMAC
    - CLASS= ‘class name’, or, class name addr
    - ACCESS=READ, READWRITE, WRITE
    - ACEE=ACEE addr
    - ACEEALET=alet addr
    - USERSECLABEL=seclabel addr
    - RESCSECLABEL=seclabel addr (resource SECLABEL)
    - LOG=NONE
    - LOGSTR=logstr addr

Misc Enhancements

- SECLABEL support for FASTAUTH
  - FASTAUTH was modified to provide support for SECLABELs
- Auditing
  - Two new Event Codes.
  - New Event Code Qualifiers and Relocate sections added to a number of events.
- Enhancements to RACF Utilities
  - In addition to the changes in UT200 and DB Unload for SERVAUTH, the following utilities have been enhanced:
    - SMF Unload
    - SAF Trace

Notices

This information was developed for products and services offered in the U.S.A.

Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of 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.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:
This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.