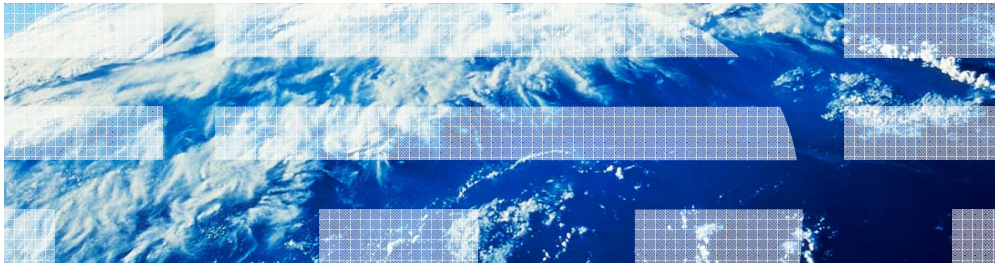


z/OS Operating System

Accessing RACF Resource Profiles through the IBM Tivoli Directory Server for z/OS



© 2011 IBM Corporation

This presentation summarizes how to access RACF resource profiles through the SDBM backend in the IBM Tivoli Directory Server for z/OS. It is applicable to customers running at z/OS 1.11 and above.

SDBM requirements

- **Problem:**
 - Customers are using SDBM to manage RACF user, group, and connect profiles
 - Want to also manage resource profiles
- **Solution:**
 - Extend SDBM operations to also cover RACF resources
 - Include support for the SETROPTS class keywords
 - No support for DATASET
 - Support logging changes to RACF resource profiles
- **Advantages:**
 - Remotely administer RACF resource profiles using SDBM

Customers have used the SDBM backend for many years to remotely manage some of the common RACF profiles - user, group, and connect profiles. They also want to be able to use the same SDBM operations to administer RACF resource profiles. SDBM can now be used to set and display resource profiles, including the RACF system settings that affect classes (such as RACLIST and REFRESH). Note that SDBM does not support managing DATASET profiles.

Customers use the LDAP change log backend, GDBM, to keep other non-z/OS directories in sync with RACF user, group, and connect profiles. The LDAP server is extended to also create change log entries in GDBM when RACF resource profiles are changed. Meta-directory applications, such as Tivoli Directory Integrator, can poll GDBM, detect a change to a RACF resource profile, use SDBM to extract the resource profile, and then propagate the profile values to other directories.

LDAP support for RACF resources overview (1 of 2)

- Extend SDBM directory hierarchy with these entries:
 - A class entry to represent each RACF class
 - DN is *profiletype=className,suffix*
 - Subtree top for resource profile entries for that class
 - A resource entry to represent each RACF resource profile
 - DN is *profilename=profileName,profiletype=className,suffix*
 - Profile name can be generic (like ABC*)
 - A single setropts entry to represent RACF system settings for classes
 - DN is *cn=setropts,suffix*

There are three types of entries that have been added to the SDBM directory to map RACF resources:

1. The SDBM backend generates a class entry to represent each RACF class (except DATASET). This includes RACF dynamically-defined classes. The class name is part of the distinguished name (DN) of the class entry. The class entry provides a collection point for the resource entries representing each RACF resource profile in the class.

2. Each RACF resource profile in the class is represented by an SDBM resource entry under the class entry. This includes both generic and discrete resource profiles. The DN of the resource entry contains both the resource profile name and the class name.

3. There are several RACF system settings that pertain to classes. These settings are encapsulated in a single setropts entry, so called because the RACF SETROPTS command is used to manage these RACF system settings. The DN of the setropts entry contains *cn=setropts* to identify the entry.

LDAP support for RACF resources overview (2 of 2)

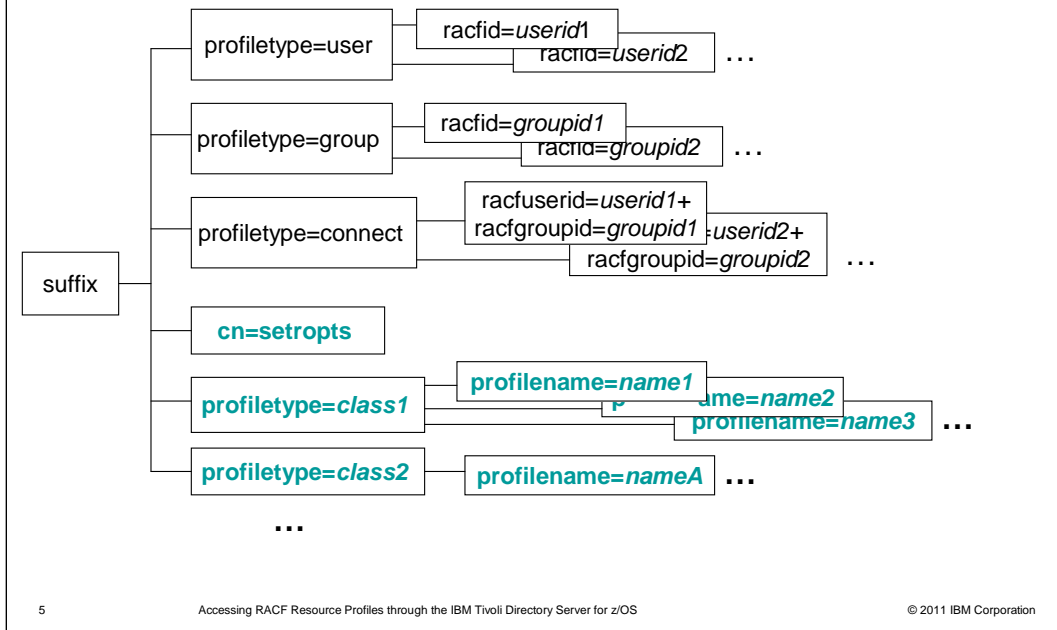
- Additional SDBM attributes and object classes in initial schema
- SDBM config option, **enableResources on**, to activate support for these entries
 - Otherwise:
 - SDBM operation fails if target is one of these entries
 - These entries not returned on SDBM searches
- DATASET class and profiles not supported, but
 - DATASET class can be attribute value in setropts entry

The SDBM entries for resource support contain attributes and object classes used to represent RACF values. These attributes and object classes are all part of the initial LDAP schema so that no additional schema is needed by SDBM to use these entries.

These SDBM entries can only be used if support for RACF resources is enabled in the LDAP server, by specifying `enableResources` on the SDBM section of the LDAP server configuration file. If this is not included in the configuration file, an SDBM operation to administer or search one of these entries fail. An SDBM search operation starting at the suffix does not return these entries even if they match the search filter.

Note that the DATASET class is not supported directly. There is no SDBM class entry for DATASET and there are no resource entries for DATASET profiles. However, the DATASET class is supported as a value in the `setropts` keywords that support classes. For example, an SDBM modify can be used to activate the DATASET class. To do this, the SDBM modify will contain something like “`racfclassact: profilename=DATASET,cn=myRacf`”.

SDBM directory hierarchy



The non-bold portions of the diagram on this slide are existing SDBM entries, used to map RACF user, group, and user-group connection profiles. The bold portions are the additional entries used to map RACF resources. There is a class entry for each class and a resource entry for each resource in each class (except the DATASET class, which is not supported). There is also a setropts entry to allow setting system values that affect classes.

SDBM class entry

- Class entry to represent each RACF class
 - DN is `profiletype=className,suffix`
 - `className` cannot be CONNECT, DATASET, GROUP, or USER, and not case sensitive
 - Top subtree entry for resource entries in that class
- Can search and compare – no add, modify, delete, rename
 - Can manage dynamic classes using resource profiles in CDT class
- Limited search support
 - Entire class entry returned on base search from class entry
 - DN-only entry returned in one-level or subtree search from suffix
 - DN-only entry returned on subtree search from entry
 - “objectclass=*” only supported filter

The class entry name cannot be DATASET as this class is not supported by the LDAP server. It also cannot be CONNECT, GROUP, or USER since those names are already in use as the top subtree entries for user, group, and connection profiles. Also note that the class name is not case sensitive.

A class entry cannot be directly added, deleted, or modified. Thus, an LDAP add of “`profiletype=myClass,cn=myRacf`” fails. Instead, a dynamic class can be managed by updating its resource profile in the CDT class. For example, an LDAP add of “`profilename=myClass,profiletype=CDT,cn=myRacf`” creates a new dynamic class named myClass.

The “objectclass=*” filter can be used to display class entries if they are in scope. When the scope includes more than one entry, then only the distinguished name (DN) of the class entry is returned. When the scope includes just a class entry (that is, the scope is base and the search base is a class entry), then the entire class entry is returned.

Class entry examples

- Display FACILITY class entry
ldapsearch ... -s base -b profiletype=facility,cn=myRacF "objectclass=""
dn: profiletype=FACILITY,cn=myRacF
objectclass: TOP
objectclass: RACFPROFILETYPE
profiletype: FACILITY
- Display all the class entries
ldapsearch ... -s one -b cn=myRacF "objectclass=""
...
dn: profiletype=ACCTNUM,cn=myRacF
dn: profiletype=ACICSPCT,cn=myRacF
...

The first example is a base search from a class entry – it returns the entire class entry. There is very little in a class entry. It is basically just a container entry in the SDBM hierarchy for the resource entries representing the resource profiles in the class.

The second example is a one-level search from the suffix. It returns all the top entries directly below the suffix – the user, group, connect subtree top entries, the class entries, and the setropts entry, although only the class entries are shown in this excerpt. Only the distinguished name (DN) of the class entries and setropts entry are returned, while the entire user, group, and connect subtree top entries are returned – this is done for compatibility with previous versions of the LDAP server.

SDBM resource entry

- An entry to represent each RACF resource profile in a class
 - DN is `profilename=profileName,profiletype=className,suffix`
 - *profileName* can be discrete or generic, case-sensitive
 - `profilename=ABC.KEN,profiletype=facility,cn=myRacf`
 - `profilename=ABC.*,profiletype=facility,cn=myRacf`
 - `profilename=Abc.Def,profiletype=ejbrole,cn=myRacf`
 - Case-sensitive in LDAP, but RACF ignores case if class not defined as MIXED
- Structural object class: **racfresource**
- Most RACF RDEFINE/RALTER keywords mapped to separate LDAP attributes
 - All segments supported except SVFMR and TME

Each RACF resource profile (except DATASET profiles) is represented by an LDAP resource entry. The distinguished name of the entry contains both the RACF resource profile name and the name of the class it is in. The resource profile name can be a RACF generic name.

The **profilename** attribute is case-sensitive to the LDAP server. For RACF classes that support case-sensitive resource names, the *profileName* case spelling in the distinguished name (DN) must exactly match the resource name. For RACF classes in which the resource names are not case-sensitive, the *profileName* casing does not matter. Thus profile name ABc is the same as ABC in the facility class, but is different in the ejbrole class (which is defined as MIXED).

The SDBM resource profile entry embodies all the keywords associated with the RACF commands and interfaces that manage and display a resource profile: RDEFINE, RALTER, RDELETE, PERMIT, and R_admin resource extract. The only segments not supported are SVFMR and TME, because they were added to RACF a while ago for products that are no longer supported. The resource profile entry uses a structural object class, **racfresource**, which just contains the **profilename** naming attribute. The **extensibleobject** object class is used to cover the rest of the attributes in the entry.

SDBM resource entry (continued)

- Two special attributes:
 - **racfResourceAttributes** - multi-valued attribute
 - Specify any one-word RACF RDEFINE/RALTER keywords (SINGLEDSN, TVTOC, WARNING, NOCDTINFO, ...)
 - **racfAccessControl** attribute to manage access control lists
 - Multi-valued - each value must be complete PERMIT command except for class name and resource name
 - SDBM issues a PERMIT command for each value
 - Examples:
 - racfaccesscontrol: id(ken) access(read)
 - racfaccesscontrol: id(XYZ) access(update) when(terminal(t1)) count(21)
 - racfaccesscontrol: from(KEN.ABC) fclass(facility)

The **racfResourceAttributes** attribute plays the same role for resource entries as the **racfAttributes** attribute for user and group entries and the **racfConnectAttributes** attribute for connect entries. This attribute provides a way to specify additional values found on the RDEFINE and RALTER commands that are not mapped to other LDAP attributes. When using a modify operation to delete specific **racfResourceAttributes** values, SDBM adds each value prepended with NO to the RALTER command. For example, deleting “TVOC” results in adding “NOTVTOC” to the RALTER command. When deleting the entire attribute, SDBM adds “NOSINGLEDSN NOTVTOC NOWARNING” to the RALTER command.

The **racfAccessControl** attribute is used to represent the keywords used in RACF PERMIT commands to manage the standard and conditional access lists for the resource profile. Each **racfAccessControl** value results in issuing a separate RACF PERMIT command. The value must contain a complete PERMIT command except for the class name and the resource name, which are added by SDBM. Also, the value can contain a COUNT field, which SDBM removes. The COUNT field can be returned by RACF on output, but is not needed on input. When using a modify operation to delete a specific **racfAccessControl** value, SDBM adds “DELETE” to the end of the PERMIT command. When deleting the entire attribute, SDBM creates a PERMIT command containing “RESET(ALL)”.

Resource entry add/modify/delete

- Can add/modify/delete resource entry – results in
 - RACF RDEFINE/RALTER/RDELETE commands for profile
 - RACF PERMIT commands for access control
 - Multiple RACF commands issued if add/modify resource attributes and access control in one LDAP operation
 - RDEFINE/RALTER followed by one or more PERMIT
 - LDAP operation fails if any command fails
 - Not atomic – previous successful RACF commands not undone if later command fails
 - Can display profile to see current state

An SDBM add, modify, or delete operation of a resource entry results in issuing the appropriate RACF commands to create, alter, or delete a RACF resource profile.

If a resource profile add or modify operation includes the **racfAccessControl** attribute, it is possible that a single LDAP add or modify operation can result in SDBM issuing multiple RACF operations. For an add operation, SDBM issues an RDEFINE command followed by a PERMIT command for each **racfAccessControl** value. For a modify operation, SDBM issues an RALTER command if some attribute other than **racfAccessControl** is specified, and then issues a PERMIT command for each **racfAccessControl** value. When issuing multiple RACF commands, the SDBM operation terminates if any of the RACF commands fail. If the failing command is not the first command, the resource profile is **NOT** restored to its original state – instead, it contains any changes made by the prior successful commands. In this case, use LDAP search to display the resource entry to see the changes that have been made.

Resource entry compare

- Can compare value in a resource entry
 - Uses RACF R_admin resource profile extract interface
 - For attributes with a class name as value, value to compare must be complete class DN
 - For example:
 - use “racfcdtinfogroup=profiletype=mygroup,cn=myracf”
 - not “racfcdtinfogroup=mygroup”
 - For **racfAccessControl**, only compares the ID part
 - Like **aclEntry** compare - determines if user is in access lists
 - Example: compare is true if “fred” is in an access list whatever access “fred” has:
 - ldapcompare ... “racfaccesscontrol=ID(fred) access(read)”

An SDBM compare operation for a resource entry uses the RACF R_admin resource profile extract interface to retrieve the resource profile from RACF and then compares the specified value.

There are several resource entry attributes which have a class name as their value. In add or modify operations, these values can be specified as a complete class entry DN or as just the class name. In a compare operation, the complete class DN must be specified. The compare is always FALSE if just the class name is used.

For the **racfAccessControl** attribute, only the ID field in the compare value is used to compare against the values in the resource profile. This is similar to how compare works for the **aclEntry** attribute: it basically answers the question “does user/group x appear in the standard or conditional access list in the resource profile?”. The rest of the compare value is ignored. In particular, a user/group might not have the ACCESS specified in the compare value (in fact, a user/group might have “ACCESS(NONE)”), but the result can still be TRUE.

Resource entry search

- Entire entry returned on base or subtree search from entry
 - “objectclass=*” is only supported filter
 - Uses RACF R_admin resource profile extract interface
- DN-only entry returned on one-level or subtree search from class entry or subtree search from suffix
 - Only supported filters: “objectclass=*” and “profilename=xxx”
 - “objectclass=*” returns DN of each resource entry in scope
 - Uses RACF SEARCH command
 - **Note: At most 4096 profile names returned per class**

An SDBM search using “objectclass=*” filter returns the entire resource entry if the base is the resource entry, using the R_admin resource profile extract interface.

If the SDBM search starts at the class entry or the suffix, it returns only the DN of each resource entry within scope. SDBM issues the “RACF SEARCH CLASS(XXX) FILTER(**)” command to get the names of the resource profiles of each class. There is a limit of 4096 resource profile names that are returned by RACF for each class, thus LDAP returns at most 4096 distinguished names (DNs) per class. There is no indication that there might be additional resource profiles that have not been returned.

Resource entry search (continued)

- “profilename=xxx” returns DNs of matching resource names
 - xxx can contain RACF search wildcards (% , * , **)
 - Note must escape ** - “profilename=**”
 - No way to limit search to match a generic profile
 - For example: “profilename=ABC.*” search returns all matches even if have a resource named ABC.*
 - xxx can be any case if class ignores case, else must match exact spelling
 - Uses RACF SEARCH command
 - **Note: At most 4096 profile names returned per class**

The “profilename=xxx” filter can be used to find the distinguished names (DNs) of resource profiles that match the filter value in a specific class or in all classes. Note that any wildcards used in a “profilename=xxx” filter are handled as actual wildcards by RACF. For example, “profilename=ABC.*” returns all matching discrete and generic resource profiles, such as ABC.X, ABC.*, and ABC.X*. There is no way to indicate that a ‘*’ in a filter is part of a generic profile name rather than a wild card. Also, “profilename=**” is not a valid LDAP filter because it has two substring wildcards in a row. To use such a filter, escape one or both of the asterisks with a backslash so that it is not treated as a substring wildcard. SDBM removes the escape characters before it uses the filter.

When searching in a class that does not support mixed case resource profile names, the profilename filter value can be any case (Abc.*, aBC.*, etc). If the class supports mixed case resource profile names, the profilename filter value must match the correct case spelling.

Again, the RACF SEARCH command only returns up to 4096 matches in each class, thus LDAP returns at most 4096 distinguished names (DNs) per class. There is no indication that there might be additional resource profiles that have not been returned.

Resource entry examples

- Display the KEN.RES profile in FACILITY class

```
ldapsearch ... -b profilename=KEN.RES,profiletype=facility,cn=myRacF  
"objectclass="
```

```
dn: profilename=KEN.RES,profiletype=FACILITY,cn=myRacF  
profilename: KEN.RES
```

```
racfauthorizationdate: 09/28/07
```

```
racfowner: RACFID=U1,PROFILETYPE=USER,CN=MYRACF
```

```
racflastreferredate: 09/28/07
```

```
racflastchangedate: 09/28/07
```

```
racfalteraccesscount: 0
```

```
racfcontrolaccesscount: 0
```

```
racfupdateaccesscount: 0
```

```
racfreadaccesscount: 0
```

```
(continued)
```

This example (continued on the next slide) shows LDAP search output for a subtree search from a resource entry. The entire entry is returned.

Resource entry examples (continued) (1 of 2)

```
racfuacc: READ
racresourceaudit: SUCCESS(UPDATE),FAILURES(READ)
racresourceglobalaudit: SUCCESS(UPDATE),FAILURES(READ)
racinstallationdata: HELLO THERE
racaccesscontrol: ID(FRED) ACCESS(UPDATE) COUNT(0)
racaccesscontrol: ID(XTAB) ACCESS(READ) COUNT(0)
racaccesscontrol: ID(X) ACCESS(UPDATE) WHEN(TERMINAL(T1)) COUNT(5)
racresourceattributes: WARNING
racsessionconvsec: NONE
objectclass: TOP
objectclass: RACFRESOURCE
objectclass: EXTENSIBLEOBJECT
```

This continuation of the resource entry search output includes the standard and conditional access lists, returned as **racfAccessControl** attribute values. The **racfAccessControl** values might contain the COUNT field. This field is ignored if the value is used as input in an SDBM add, modify, or compare operation.

Note that there is a structural object class, **racfResource**, for resource entries. The object class contains the naming attribute, **profilename**. The **extensibleObject** object class is used to cover all the other attributes in the entry.

Resource entry examples (continued) (2 of 2)

- Display DNs of all resource profiles whose name matches KEN.*
ldapsearch ... -b cn=myRacF "profilename=KEN.*"

```
dn: profilename=KEN.*,profiletype=\#X#Y,cn=myRacF
```

```
dn: profilename=KEN.*,profiletype=\#X#Y,cn=myRacF
```

```
dn: profilename=KEN.%A,profiletype=\#X#Y,cn=myRacF
```

```
dn: profilename=KEN.FRED,profiletype=\#X#Y,cn=myRacF
```

```
dn: profilename=KEN.ABC,profiletype=FACILITY,cn=myRacF
```

```
dn: profilename=KEN.*,profiletype=FACILITY,cn=myRacF
```

The profilename filter is used to find resource entries whose name matches the profilename filter value. Note that the wildcards used in “profilename=xxx” filter are always handled as actual wildcards by RACF. For example, “profilename=KEN.*” returns distinguished names (DNs) of both discrete and generic resource profiles whose name matches KEN.*. There is no way to use the profilename filter with a wildcard to get an exact match only on a generic resource profile (that is, to just match KEN.*).

A pound sign (#) in a DN is escaped with a backslash because pound signs have special meaning in LDAP. Thus, the #X#Y class is represented in a DN as profiletype=\#X\#Y.

SDBM Setropts entry

- Single entry to represent RACF system settings for classes
 - DN is `cn=setropts,suffix`
- Most class-oriented keywords of RACF SETROPTS command mapped to separate LDAP attributes
 - AUDIT, CLASSACT, GENCMD, GENERIC, GENLIST, GLOBAL, LOGOPTIONS, RACLIST, STATISTICS
- Can use special **racfSetroptsAttributes** attribute to specify REFRESH and WHEN(PROGRAM)
- Cannot specify any other SETROPTS keywords

There is a single SDBM setropts entry to set and display the RACF system settings that pertain to classes. These system settings are managed using the RACF SETROPTS command. The class-oriented keywords in the RACF SETROPTS command are mapped to their own LDAP attributes except for REFRESH, WHEN, and NOWHEN. These values can be specified on the special **racfSetroptsAttributes** attribute. If a modify operation deletes the entire **racfSetroptsAttribute**, then "NOWHEN(PROGRAM)" is added to the RACF SETROPTS command. The other keywords in the RACF SETROPTS command cannot be specified within the SDBM setropts entry.

Setropts entry operations

- Can modify, search, compare – no add, delete, rename
 - Modify invokes RACF SETROPTS command
 - Search/compare invoke R_admin setropts extract
- Modify input - class name can be complete DN or just name
 - racfraclist: profiletype=cdt,cn=myRac
 - racfraclist: facility
- Compare value: must use complete DN, not just name
 - ldapcompare ... "racfraclist=profiletype=cdt,cn=myRac"
- Can specify DATASET class DN in attribute value
 - racfglobal: profiletype=dataset,cn=myRac

The SDBM modify, search and compare operations can be used with setropts entry. The setropts entry always exists when SDBM resource support is enabled – the entry cannot be added, deleted, or renamed. Modifying the setropts entry results in issuing the RACF SETROPTS command. Displaying or comparing the setropts entry results in using the R_admin setropts extract interface to retrieve the contents of the setropts options.

The values of most of the attributes used in the setropts entry represent classes. On input to a modify operation, the value can be specified as a complete class distinguished name (DN) or as just the class name. For a compare operation, the value to compare must be a complete class DN. The compare always returns FALSE if the compare value is just a class name.

Although there is no DATASET class entry and the dataset DN (profiletype=dataset,*suffix*) cannot be the target of an SDBM operation, the dataset DN can be specified as an attribute value when modifying and comparing the setropts entry.

Setropts entry operations (continued)

- Search support
 - Entire entry returned in base or subtree search from setropts entry
 - DN-only entry returned in one-level or subtree search from suffix
 - “objectclass=*” only supported filter
- Search output – class names returned as complete DNs (including DATASET class DN)

The only supported filter for the setropts entry is “objectclass=*”. The entire entry is returned if the base of the search is the setropts entry and the scope is base or subtree. For a one-level or subtree search from the suffix, only the distinguished name (DN) of the setropts entry is returned.

When displaying the setropts entry, the values that are class names are returned as complete class DNs. The DATASET class DN is returned if it is one of the values.

Setropts entry examples

- Display setropts entry

```
ldapsearch ... -b cn=setropts,cn=myRacF "objectclass=*"
dn: cn=setropts,cn=myRacF
objectclass: TOP
objectclass: CONTAINER
objectclass: EXTENSIBLEOBJECT
cn: setropts
racfclassact: PROFILETYPE=DATASET,CN=MYRACF
racfclassact: PROFILETYPE=USER,CN=MYRACF
racfclassact: PROFILETYPE=GROUP,CN=MYRACF
racfclassact: PROFILETYPE=ACCTNUM,CN=MYRACF
...
(continued)
```

This example (continued on the next slide) shows an excerpt of the setropts entry retrieved by an SDBM subtree search targeted to the setropts entry. The values returned for attributes such as **racfclassact** are complete class DNs. The DATASET class DN is returned because it is one of the values.

Note that the **container** and **extensibleobject** object classes are used to cover the attributes used in the setropts entry.

Setropts entry examples (continued)

```
...
racfgencmd: PROFILETYPE=DATASET,CN=MYRACF
racfgencmd: PROFILETYPE=ACCTNUM,CN=MYRACF
...
racfraclist: PROFILETYPE=APPCTP,CN=MYRACF
racfraclist: PROFILETYPE=APPL,CN=MYRACF
racfraclist: PROFILETYPE=CDT,CN=MYRACF
...
racflogoptionsdefault=PROFILETYPE=DATASET,CN=MYRACF
racflogoptionsdefault=PROFILETYPE=\\#X\\#Y,CN=MYRACF
...
racflogoptionsdefault: PROFILETYPE=X1X,CN=MYRACF
racfsetroptsattributes: NOWHEN(PROGRAM)
```

This is a continuation of the excerpt of SDBM search output for the setropts entry. A pound sign (#) contained in a DN is escaped with a backslash because pound signs have special meaning in LDAP.



Feedback

Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send email feedback:

mailto:iea@us.ibm.com?subject=Feedback_about_V1R11_RACF_SDBM.ppt

This module is also available in PDF format at: [../V1R11_RACF_SDBM.pdf](..\\V1R11_RACF_SDBM.pdf)

You can help improve the quality of IBM Education Assistant content by providing feedback.



Trademarks, disclaimer, and copyright information

IBM, the IBM logo, ibm.com, RACF, Tivoli, and z/OS are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of other IBM trademarks is available on the web at "[Copyright and trademark information](http://www.ibm.com/legal/copytrade.shtml)" at <http://www.ibm.com/legal/copytrade.shtml>

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS OR SOFTWARE.

© Copyright International Business Machines Corporation 2010. All rights reserved.