Db2 11 for z/OS

RACF Access Control Module Guide

SC19-4065-06

Notes

Before using this information and the product it supports, be sure to read the general information under "Notices" at the end of this information.

Subsequent editions of this PDF will not be delivered in IBM Publications Center. Always download the latest edition from PDF format manuals for Db2 11 for z/OS (Db2 for z/OS in IBM Documentation).

2022-01-14 edition

This edition applies to Db2® 11 for z/OS® (product number 5615-DB2), Db2 11 for z/OS Value Unit Edition (product number 5697-P43), and to any subsequent releases until otherwise indicated in new editions. Make sure you are using the correct edition for the level of the product.

Specific changes are indicated by a vertical bar to the left of a change. A vertical bar to the left of a figure caption indicates that the figure has changed. Editorial changes that have no technical significance are not noted.

Contents

About this information

This information describes planning, installing, and implementing the RACF® access control module, a sample exit routine that ships with Db2 for z/OS.

Throughout this information, "Db2" means "Db2 11 for z/OS". References to other Db2 products use complete names or specific abbreviations.

Important: To find the most up to date content for Db2 11 for z/OS, always use IBM® Documentation or download the latest PDF file from PDF format manuals for Db2 11 for z/OS (Db2 for z/OS in IBM Documentation).

This information assumes that Db2 11 is running in new-function mode, and that your application is running with the application compatibility value of 'V11R1', except for the following section that describe the migration process and how to activate new function:

- Migrating to Db2 11 (Db2 Installation and Migration)
- What's new in Db2 11 (Db2 for z/OS What's New?)
- Changes in Db2 11 (Db2 for z/OS What's New?)

Availability of new function in Db2 11

The behavior of data definition statements such as CREATE, ALTER, and DROP, which embed data manipulation SQL statements that contain new capabilities, depends on the application compatibility value that is in effect for the application. An application compatibility value of 'V11R1' must be in effect for applications to use new capability in embedded statements such as SELECT, INSERT, UPDATE, DELETE, MERGE, CALL, and SET *assignment-statement*. Otherwise, an application compatibility value of 'V10R1' can be used for data definition statements.

Generally, new SQL capabilities, including changes to existing language elements, functions, data manipulation statements, and limits, are available only in new-function mode with applications set to an application compatibility value of 'V11R1'.

Optimization and virtual storage enhancements are available in conversion mode unless stated otherwise.

SQL statements can continue to run with the same expected behavior as in DB2® 10 new-function mode with an application compatibility value of 'V10R1'.

Who should read this information

Use this information as a guide to the task of planning, installing, and implementing the RACF access control module. The skills required include MVS™ system programming, Db2 administration, and RACF administration. The participants for this task should include those who are knowledgeable in the current security structure and policies in place for both Db2 and RACF at your installation.

Db2 Utilities Suite for z/OS

Important: In Db2 11, the Db2 Utilities Suite for z/OS is available as an optional product. You must separately order and purchase a license to such utilities, and discussion of those utility functions in this publication is not intended to otherwise imply that you have a license to them.

Db2 11 utilities can use the DFSORT program regardless of whether you purchased a license for DFSORT on your system. For more information, see the following informational APARs:

- II14047
- II14213
- II13495

Db2 utilities can use IBM Db2 Sort for z/OS (5655-W42) as an alternative to DFSORT for utility SORT and MERGE functions. Use of Db2 Sort for z/OS requires the purchase of a Db2 Sort for z/OS license. For more information about Db2 Sort for z/OS, see Db2 Sort for z/OS.

Related concepts

Db2 utilities packaging (Db2 Utilities)

Terminology and citations

When referring to a Db2 product other than Db2 for z/OS, this information uses the product's full name to avoid ambiguity.

The following terms are used as indicated:

Db2
Represents either the Db2 licensed program or a particular Db2 subsystem.

IBM re-branded DB2 to Db2, and Db2 for z/OS is the new name of the offering previously know as "DB2 for z/OS". For more information, see Revised naming for IBM Db2 family products on IBM z/OS platform. As a result, you might sometimes still see references to the original names, such as "DB2 for z/OS" and "DB2", in different IBM web pages and documents. If the PID, Entitlement Entity, version, modification, and release information match, assume that they refer to the same product.

Tivoli® OMEGAMON® XE for Db2 Performance Expert on z/OS
Refers to any of the following products:

- IBM Tivoli OMEGAMON XE for Db2 Performance Expert on z/OS
- IBM Db2 Performance Monitor on z/OS
- IBM Db2 Performance Expert for Multiplatforms and Workgroups
- IBM Db2 Buffer Pool Analyzer for z/OS

C, C++, and C language
Represent the C or C++ programming language.

CICS®
Represents CICS Transaction Server for z/OS.

IMS
Represents the IMS Database Manager or IMS Transaction Manager.

MVS
Represents the MVS element of the z/OS operating system, which is equivalent to the Base Control Program (BCP) component of the z/OS operating system.

RACF
Represents the functions that are provided by the RACF component of the z/OS Security Server.

Accessibility features for Db2 11 for z/OS

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use information technology products successfully.

Accessibility features

The following list includes the major accessibility features in z/OS products, including Db2 11 for z/OS. These features support:

- Keyboard-only operation.
- Interfaces that are commonly used by screen readers and screen magnifiers.
- Customization of display attributes such as color, contrast, and font size

Tip: IBM Documentation (which includes information for Db2 for z/OS) and its related publications are accessibility-enabled for the IBM Home Page Reader. You can operate all features using the keyboard instead of the mouse.

Keyboard navigation

For information about navigating the Db2 for z/OS ISPF panels using TSO/E or ISPF, refer to the *z/OS TSO/E Primer*, the *z/OS TSO/E User's Guide*, and the *z/OS ISPF User's Guide*. These guides describe how to navigate each interface, including the use of keyboard shortcuts or function keys (PF keys). Each guide includes the default settings for the PF keys and explains how to modify their functions.

Related accessibility information

IBM and accessibility

See the *IBM Accessibility Center* at http://www.ibm.com/able for more information about the commitment that IBM has to accessibility.

How to send your comments about Db2 for z/OS documentation

Your feedback helps IBM to provide quality documentation.

End of support (EOS): Db2 11 reached EOS on March 31, 2021. The online product documentation is provided as-is for clients with extended service contracts. For more information, see End of support (March 31, 2021) (Db2 for z/OS in IBM Documentation).

Send any comments about Db2 for z/OS and related product documentation by email to db2zinfo@us.ibm.com.

To help us respond to your comment, include the following information in your email:

- The product name and version
- The address (URL) of the page, for comments about online documentation
- The book name and publication date, for comments about PDF manuals
- The topic or section title
- The specific text that you are commenting about and your comment

Related concepts

About this information (Db2 for z/OS in IBM Documentation)

Related reference

PDF format manuals for Db2 11 for z/OS (Db2 for z/OS in IBM Documentation)

Chapter 1. Introduction to the RACF access control module

The RACF access control module allows you to use RACF in addition to Db2 authorization checking for Db2 objects, authorities, commands, and utilities.

You can activate the RACF access control module at the Db2 access control authorization exit point (DSNX@XAC), where you can replace the default Db2 exit routine.

The RACF access control module:

- Receives control from the Db2 access control authorization exit point (DSNX@XAC) to handle Db2 authorization checks
- Provides a single point of control for RACF and Db2 security administration
- Provides the ability to define security rules before a Db2 object is created
- Allows security rules to persist when a Db2 object is dropped
- Provides the ability to protect multiple Db2 objects with a single security rule using a combination of RACF generic, grouping, and member profiles
- Eliminates revocation of dependent privileges when a privilege is revoked from a Db2 user.
- Preserves Db2 privileges and administrative authorities
- Provides flexibility for multiple Db2 subsystems with a single set of RACF profiles
- Allows you to validate a user ID before giving it access to a Db2 object.

RACF support for the RACF access control module includes a set of general resource classes in the RACF module ICHRRCDX (the supplied portion of the RACF class descriptor table). These classes are used when you implement the RACF access control module using the default values.

RACF checking for Db2 resources

Each Db2 command, utility, and Structure Query Language (SQL) statement is associated with a set of privileges, authorities, or both.

Authority checking performed by the RACF access control module simulates Db2 authority checking:

- Db2 object types map to RACF class names
- Db2 privileges map to RACF resource names for Db2 objects
- Db2 authorities map to the RACF administrative authority class (DSNADM) and RACF resource names for Db2 authorities
- Db2 security rules map to RACF profiles

The RACF access control module checks the RACF profiles corresponding to that set of privileges and authorities.

See Chapter 10, “Special considerations,” on page 43 and Chapter 15, “RACF authorization checking reference,” on page 83 for more information.

Multilevel security

You can improve the security of your Db2 applications when you add RACF security labels to Db2 objects or row-level security on a multilevel-secure system.

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.

This document does not address the use of Db2 and the RACF access control module in a multilevel-secure environment.

Related reference

z/OS multilevel security and the Common Criteria (Planning for Multilevel Security and the Common Criteria)

The Db2 access control authorization exit point

Db2 provides an exit point so you can install the RACF access control module.

If you install the RACF access control module, RACF can perform Db2 authorization checking for SQL statements, commands, and utilities. You can also choose to provide your own routine for the Db2 access control authorization exit point. This document describes how to implement only the supplied RACF access control module and DB2 access control authorization exit.

Related concepts

Access control authorization exit routine (Managing Security)

The default Db2 exit routine

The default Db2 exit routine at the DSNX@XAC exit point returns a code to the Db2 authorization module.

The code indicates that an installation-defined access control authorization exit routine is unavailable. Db2 then performs native authorization checking and does not attempt to invoke this exit routine again. The default Db2 exit routine called DSNX@XAC is in library *prefix*.SDSNLOAD. The source code for the default Db2 exit routine is in the DSNXSXAC member of *prefix*.SDSNSAMP. The Db2 installation process puts the results of the assembly into *prefix*.SDSNEXIT.

By contrast, the RACF access control module is provided in DSNXRXAC member of *prefix*.SDSNSAMP and provides access control using a combination of RACF and Db2 checking. You can easily alter the Db2 installation process by modifying the DSNTIJEX job to assemble the RACF access control module, rather than the default Db2 exit routine.

When the RACF access control module is invoked

The RACF access control module is invoked when Db2 starts, shuts down, or when authorization checking is performed for a privilege.

The RACF access control module is invoked in three instances:

- At Db2 startup

  When Db2 starts, the RACF access control module is invoked to allow the external authorization checking application to perform any required setup prior to authorization checking. An example of a required setup task is loading authorization profiles into storage. Db2 uses the reason code that the exit routine sets during startup to determine how to handle exception situations.
- When an authorization check is to be performed for a privilege

  At the point when Db2 would access security tables in the catalog, to check authorization on a privilege, the RACF access control module is invoked. The exit routine is only invoked if none of the prior invocations have indicated that the exit routine must not be called again.
- At Db2 shutdown

  When Db2 is stopping, the RACF access control module is invoked to let the external authorization checking application perform its cleanup before Db2 stops.

When the RACF access control module is bypassed

RACF access control module is not always called to check authorization.

In the following situations, the RACF access control module is not called to check authorization:

- The user has installation SYSOPR (when sufficient for the privilege being checked) or installation SYSADM authority. This authorization check is made strictly within Db2.

  The RACF access control module is called for any additional authorization checks that are done as part of a process, if those checks are done on behalf of another user or role that does not have installation SYSADM or installation SYSOPR authority. An example of such a process is revoking of dependent privileges.
- Db2 security has been disabled (NO was specified in the USE PROTECTION field of installation panel DSNTIPP).
- Db2 cached the authorization information from a prior check.
- From a prior invocation of the RACF access control module, the routine had indicated that it should not be called again.
- Db2 GRANT statements are issued to control authorization by granting privileges in Db2.

Chapter 2. Planning

You must develop a plan with your team members before you implement the RACF access control module.

Implementing the RACF access control module involves the interaction of RACF, Db2 and z/OS system software, each with its own required skills. Therefore, it is important to understand the task at hand, organize the appropriate team members, and plan your implementation together.

This chapter provides the information you must determine the tasks to be performed, identify the skills required, recognize decisions that you make as a team, and understand how each choice affects Db2 authorization processing.

Mapping out the implementation tasks: A task roadmap

You must make important decisions during planning that affect the RACF access control module.

The following table shows the subtasks, participants, and associated procedures for implementing the RACF access control module.

Before you begin: Important decisions that you make during planning (Subtask 1) are implemented during Subtasks 2–5.

Table 1. Task roadmap for implementing the RACF access control module

Subtask	Participants	Associated procedure
1. Plan your RACF access control module implementation.	Db2 administrator RACF administrator	See Chapter 2, “Planning,” on page 5.
2. Install and customize the RACF access control module.	MVS programmer	See Chapter 3, “Installing the RACF access control module,” on page 13.
3. (Optional) Define RACF classes for your Db2 resources, such as Db2 objects and administrative authorities.	MVS programmer	See Chapter 4, “Defining classes for the RACF access control module,” on page 17.
4. Define RACF resources to protect your Db2 objects.	RACF administrator	See Chapter 5, “Protecting Db2 objects,” on page 23.
5. Define RACF resources to protect the Db2 administrative authorities.	RACF administrator	See Chapter 6, “Protecting Db2 administrative authorities,” on page 27.
6. (Optional) If you plan to use Db2 roles, define RACF profiles to authorize users to the appropriate RACF-protected resources when they are using a role.	RACF administrator	See “Setting up profiles for Db2 roles” on page 54.
7. Activate the RACF classes for your Db2 resources and administrative authorities.	RACF administrator	See Chapter 7, “Making your new RACF resources effective,” on page 29.
8. Restart the Db2 subsystem.	Db2 administrator	—

Identifying skill requirements

You can control authorization based on user skills and the tasks that users must perform.

Organizing your team involves incorporating various skill sets and might require you to include people from different disciplines if you work in a large organization. These skills are identified in terms of the roles or job titles of the people who specialize in those skills. For example, a task requiring MVS system skills is referred to as a task for the MVS programmer. If some of your team members have multiple skills, you might require fewer individuals to complete your team.

Your team for planning and implementing the RACF access control module must include the following members:

- MVS programmer
- RACF administrator
- Db2 administrator.

The following table lists the team members, tasks, and required skills for planning and implementing the RACF access control module.

Table 2. Roles, tasks, and skills for the implementation team

Role	Tasks	Required skills	Useful references
MVS programmer	• Install (customize, assemble, and link edit) the RACF access control module • Define the RACF classes for use with Db2	• TSO skills • JCL knowledge • Assembler programming	• Security Server RACF Macros and Interfaces • Methods for associating started procedures with RACF identities (z/OS Security Server RACF System Programmer's Guide) • z/OS multilevel security and the Common Criteria (Planning for Multilevel Security and the Common Criteria) • Installing or migrating to Db2 11 (Db2 Installation and Migration)
RACF administrator	• Plan RACF classes for use with Db2 • Define RACF resources to protect Db2 objects and administrative authorities • Activate the RACF classes for Db2	• RACF administration • RACF commands, such as the following: – ADDGROUP – ADDUSER – RALTER – RDEFINE – PERMIT – SETROPTS • TSO skills	• RACF security administration (Security Server RACF Security Administrator's Guide) • z/OS Security Server RACF Command Language Reference • (optional) z/OS multilevel security and the Common Criteria (Planning for Multilevel Security and the Common Criteria)

Table 2. Roles, tasks, and skills for the implementation team (continued)

Role	Tasks	Required skills	Useful references
Db2 administrator	• Plan the Db2 objects and administrative authorities to protect • Restart the Db2 subsystem	• Db2 basic operations • Db2 commands and authorization requirements • System and basic database administration	• Db2 concepts (Db2 SQL) • Introduction to Db2 data sharing (Db2 Data Sharing Planning and Administration)

Planning for conversion

You can encounter two types of conversions when you install the RACF access control module, which is supplied in the DSNXRXAC member of prefix.SDSNSAMP.

One type of conversion involves converting from Db2 internal security, where you do not use RACF for access control authorization to Db2 resources. The other involves converting from a previous level of the RACF access control module, where you are already using RACF for access control authorization to Db2 resources.

Converting from Db2 internal security

When you convert from Db2 internal security to the RACF access control module, you do not need to convert protection for every Db2 object.

You can begin using the RACF access control module before defining profiles to protect all Db2 object types. Consider adding the WARNING option of RDEFINE and RALTER commands when you protect Db2 objects. The use of warnings might ease your conversion by allowing you to see ICH408I messages that identify profiles that would fail a request.

Any request to access a Db2 object protected by a RACF profile with the WARNING option is always allowed. If the request would have failed without the WARNING option, an ICH408I message is generated to identify the first profile (in the sequence of RACF authorization checking) that would have failed the request.

Note: When the WARNING option is added to a resource requested by a user with a Db2 administrative authority, such as SYSADM, DBADM, or in some cases SYSCTRL, that would also allow the user to access the object, you can ignore the warning message.

If the RACF access control module determines that there is no administrative authority profile and no profile to protect a particular Db2 object (or the class corresponding to a particular Db2 resource is not active), it defers to Db2 for authority checking.

For example, suppose only the set of RACF profiles to protect Db2 tables has been defined and the classes for all other object types have not been made active. In this case, the RACF access control module performs profile checking for Db2 tables, views, and indexes and it defers to Db2 for authority checking of other object types, such as plans, packages, and databases.

Guideline: All Db2 administrative authorities should be defined with UACC(NONE) before you activate the RACF access control module. You can then selectively authorize specific users at a higher level by executing the PERMIT command.

Sharing the RACF database

During migration to a new version of Db2 for z/OS, you can share the RACF database with different versions of Db2 subsystems.

Choosing the RACF access control module customization options

When you modify the customization options from their default values, you can define classes in the installation-supplied class descriptor table.

Using the default values allows the RACF access control module to use the classes in the class descriptor table (CDT) supplied by IBM. (See Chapter 13, “Supplied RACF resource classes for Db2,” on page 73.)

The RACF access control module uses the values &CLASSOPT, &CLASSNMT, and &CHAROPT to determine the format of the class names and resource names it constructs to protect the Db2 objects. The decisions you make about changing or keeping these defaults should be well understood before you complete “Installing the RACF access control module” on page 13.

Restriction: Each option that you specify in the RACF access control module applies to the entire Db2 subsystem using the module. This means that the &CLASSOPT, &CLASSNMT, and &CHAROPT values you select apply to all classes used by that Db2 subsystem. If you have multiple Db2 subsystems and must apply different values across subsystems, install the RACF access control module separately on each subsystem, each with its own set of processing options.

Table 3. Set symbols and values

Set symbol	Description	Default value	See...
&CLASSOPT	Specifies the class scope option. **Valid values:** **1** Single-subsystem scope **2** Multiple-subsystem scope	**2**	“Choosing the class scope” on page 9
&CLASSNMT	Specifies the class name *root*, which is characters 2–5 of the class name, and is used only when you also specify &CLASSOPT 2. (When you specify &CLASSOPT 1, the Db2 subsystem name or, if data sharing, the Db2 group attachment name, is used as the class name root.) **Rule:** This value must be 1–4 characters long.	**DSN**	“Choosing the class name root and suffix” on page 10
&CHAROPT	Specifies the class name *suffix*, which is the last character of the class name for installation-defined classes. **Valid values:** 0–9, #, @, $, or a blank character (' ').	**1**	“Choosing the class name root and suffix” on page 10
&ERROROPT	Specifies the action to take in the event of an initialization or authorization error. **Valid values:** **1** Native Db2 authorization is used. This is the default. **2** The Db2 subsystem is requested to stop.	**1**	“Choosing the error option” on page 10
&PCELLCT	Specifies the number of primary work area cells	**50**	“Customizing the number of exit work area cells” on page 10
&SCELLCT	Specifies the number of secondary work area cells	**50**	“Customizing the number of exit work area cells” on page 10
&SERVICELEVEL	For IBM use only		

The default values for all customization options as shipped with the RACF access control module are shown in the following figure.

```
       GBLC  &CLASSNMT,&CHAROPT,&CLASSOPT
       GBLA  &PCELLCT,&SCELLCT
&CLASSOPT     SETC  '2'     1 - Use Single Subsystem Class Scope
*                               Classification Model I
*                               (One set of classes for EACH subsys)
*                           2 - Use Multi-Subsystem Class Scope
*                               Classification Model II
*                               (One set of classes for ALL subsys)
&CLASSNMT     SETC  'DSN'   DB2 Subsystem Name (Up to 4 chars)
&CHAROPT      SETC  '1'     One character suffix (0-9, #, @ or $)
&ERROROPT     SETC  '1'     1 - Use Native DB2 authorization
*                           2 - Stop the DB2 subsystem
&PCELLCT      SETA  50      Primary Cell Count
&SCELLCT      SETA  50      Secondary Cell Count
```

Figure 1. Default values for installation options

Choosing the class scope

The system programmer can select the scope for the Db2 classes that protect Db2 objects and privileges.

The system programmer can alter the &CLASSOPT field of the modifiable assembler source statement in the RACF access control module to select the scope for the Db2 classes that will protect Db2 objects and privileges.

&CLASSOPT value	Scope	Classification model
1	Single-subsystem scope	1
2	Multiple-subsystem scope **Note:** *This is the default.*	2

When you select *single-subsystem scope*, you are choosing to define a separate set of classes for each Db2 subsystem that uses the RACF access control module. In general, you cannot use the classes in the supplied class descriptor table (ICHRRCDX) in single-subsystem scope.

When you select the *multiple-subsystem scope*, you are choosing to share a set of classes across all Db2 subsystems using RACF access control module, rather than defining a separate set for each. In multiple-subsystem scope, you can use the classes in the supplied class descriptor table (ICHRRCDX). This scope generally requires less administrative effort to set up and is the scope that most installations choose.

One general resource class is associated with each Db2 object type. You can define up to two classes for each object type and set them up as associated members or grouping classes. The list of supported Db2 objects and class abbreviations is defined in “Db2 object types” on page 23. If only one class is used for an object, it must be defined with the member prefix. An additional class is used to support Db2 administrative authorities. The format of the class names of Db2 objects depends on the classification model you use.

System considerations

When you choose single-subsystem scope and need to add a new Db2 subsystem or upgrade the RACF access control module to support a new Db2 object type, you must add new RACF classes to the RACF class descriptor table.

Tip: Add the classes to the dynamic class descriptor table so that you don't need to re-IPL your system.

When you choose multiple-subsystem scope, you can dynamically define new RACF resources to protect Db2 objects using existing classes.

If you define new RACF resources to protect Db2 objects in a class that was not active at the time your Db2 subsystem was started, you need to restart the Db2 subsystem to activate the new resources. If

the class was active at startup time, then you can dynamically activate the new resources using the TSO **SETROPTS RACLIST REFRESH** command for the class.

Related tasks

Making your new RACF resources effective
You must take several steps to ensure that your new resource definitions are effective.

Related reference

RACF security administration (Security Server RACF Security Administrator's Guide)

Choosing the class name root and suffix

The system programmer can alter the default naming conventions for the resource classes and profiles that protect Db2 objects and administrative authorities.

Once a class scope is selected, the system programmer can use the &CHAROPT and &CLASSNMT SET symbols to alter the default naming conventions for the resource classes and profiles you use to protect Db2 objects and administrative authorities.

Choosing the error option

You can specify an action for your system to take in the event of an initialization or authorization error.

Set the &ERROROPT value to choose which action you want the system to take in the event of an initialization or authorization error. If you do not set this option or allow it to default to &ERROROPT 1, native Db2 authorization is used in the event of an error.

If you select &ERROROPT 2, you can request the Db2 subsystem to stop when one of the following events occurs:

- An initialization error, such as when there are no active RACF classes found for the initializing Db2 subsystem.
- The exit routine abends, causing the accumulated number of exit routine abends to exceed the threshold specified during installation (AUTH EXIT LIMIT).
- Db2 receives an unexpected return code (EXPLRC1) from the RACF access control module.

Customizing the number of exit work area cells

When you invoke the RACF access control module, it uses CPOOL cells as a work area to contain local variables.

When you invoke the RACF access control module for initialization, it allocates a primary pool of work area cells to be used on authorization requests. Each time the RACF access control module is invoked for an authorization request, it obtains a cell and returns it when processing completes. If there are no more cells available, it uses a secondary pool of cells. You can control the number of cells allocated in the primary and secondary cell pools with the &PCELLCT and &SCELLCT SET symbols.

Guideline: Use the &PCELLCT and &SCELLCT default values.

Planning RACF security for Db2

The most significant part of the planning process is planning to expand RACF protection and administration to Db2 subsystem resources.

Plan to cover the following tasks.

1. Examining the current RACF environment, including the user group structure, resource naming conventions, and use of grouping classes.
2. Examining the Db2 objects, looking for naming conventions and other similarities in resource names that you can exploit with generic RACF profiles.

3. Examining the GRANT authorizations in place for Db2 objects to see which RACF user groups you can define, or exploit, to reduce the RACF authorizations you must create using the RACF PERMIT command.
4. Planning which Db2 objects and administrative authorities to protect, determining access requirements, and incorporating the new subsystem resources into the current RACF structure.
5. Considering the use of RACF variables to facilitate resource naming conventions for Db2 resources.
6. Integrating new Db2 users into the RACF user structure and delegating RACF group and class authorities.

Chapter 3. Installing the RACF access control module

Before your installation can use RACF to protect Db2 objects and authorities, you must install the RACF access control module.

About this task

The RACF access control module is an assembler source module that resides in the DSNXRXAC member of the *prefix*.SDSNSAMP library. To install the RACF access control module for a Db2 subsystem, you will copy, customize as needed, assemble, and link edit the module into the Db2 exit library (*prefix*.SDSNEXIT).

You can modify the way the RACF access control module works by customizing several assembler SET symbols located in the top of the source data set. The default values for all customization options as shipped with the RACF access control module are shown in "Choosing the RACF access control module customization options" on page 8.

Multiple Db2 subsystems can share the same copy of the RACF access control module as long as they use the same customization options. When subsystems require different options, you must install additional copies of the RACF access control module. Be sure that you associate each module with the correct Db2 version.

After you install the RACF access control module, it will become active the next time the Db2 subsystem is restarted when at least one RACF class associated with the Db2 subsystem is active at the time of the restart. Before restarting Db2, be sure that your implementation team has already defined appropriate RACF resources in the active Db2 classes or else your installation might cause unintended Db2 authorization failures or exposures.

Installing the RACF access control module

You can install the RACF access control module so that Db2 starts RACF for authority checking.

Before you begin

Before you install the RACF access control module, you must meet the following prerequisites:

- You must have MVS system programming skills to complete this procedure.
- In Step 3, you can optionally customize the RACF access control module to modify several important authorization processing options. Consult your implementation team to find out which customization options are needed, if any.
- You might want to have Installing or migrating to Db2 11 (Db2 Installation and Migration) available as a reference.

Procedure

To install the RACF access control module:

1. Locate the DSNXRXAC member (containing the RACF access control module) in the *prefix*.SDSNSAMP library and copy it to a private library.
2. Optionally, customize your private copy of the RACF access control module by modifying the assembler SET options from their default values. The options you use in this step affect Db2 authorization processing so use the values chosen by your implementation team.
3. Use the Db2 installation job to assemble and link edit the APF-authorized Db2 exit load library (*prefix*.). If you use another target library, you might have to change the STEPLIB or JOBLIB concatenations in the Db2 startup procedures.

a) Modify Step 3 (JEX0003) of DSNTIJEX to point to the library containing your customized version of DSNXRXAC and then run it.

b) If you have two or more Db2 subsystems and you want to use different assembler SET options for each subsystem (or you want to have separate exit load libraries), repeat the previous step for each Db2 subsystem.

Results

After you complete these steps, the RACF access control module will be initialized the next time the Db2 subsystem is started. The initialization function is successful and the RACF access control module becomes active only if Db2 resource classes are active at the time of the restart. If the RACF access control module is active, Db2 invokes RACF for authority checking.

You can determine whether Db2 performs Db2 authorization checks by reviewing the IRR9*nnx* messages and any DSNX210I message you receive during Db2 initialization.

If you receive the IRR912I message during initialization, your exit routine is not active and native Db2 authorization checking is used.

Related concepts

Choosing the RACF access control module customization options
When you modify the customization options from their default values, you can define classes in the installation-supplied class descriptor table.

Testing that your exit routine is active

You can test if your exit routine is active by causing an authorization failure.

About this task

When you complete this test, you will know if RACF is performing Db2 authorization checking. If it is, the RACF access control module is active.

Also, you might check the Db2 trace facility. The Db2 trace record IFCID 314 is only generated when the RACF access control module is active.

Procedure

To test if your exit routine is active:

1. Choose a RACF-defined Db2 table on which to execute a SELECT statement and choose an authorization ID to run the SELECT statement.

   The authorization ID must *not* own the table and have *none* of the following access authorizations:

   - Db2 administrative authority (installation SYSADM, SYSADM, SYSCTRL, or DBADM for the database containing the table. If the table is in an implicitly created database, DBADM should not be held on DSNDB04.)
   - Db2 SELECT privilege on the chosen table
   - RACF authorization for the SELECT privilege on the chosen table
   - RACF authorization for READ access to the chosen table
2. Use the authorization ID to run a SELECT statement on the table.

   The SELECT statement should fail.
3. Review the resulting ICH408I information messages related to Db2 resources and examine the RACF return code.

RACF informational messages

You can use informational messages to see how RACF is set up for a particular subsystem.

After you successfully activate the RACF access control module and Db2 invokes RACF for authorization checking, you can use the information found in messages IRR908I through IRR911I and IRR916I to see how RACF is set up for a particular subsystem.

These messages identify:

- The Db2 subsystem name, or in a Db2 data sharing environment, the Db2 group attachment name
- The FMID of the RACF access control module () or APAR number associated with the module
- The length of the RACF access control module
- The options used for the module

  For example, &ERR0R0PT specifies the correct action to be taken for Db2 initialization and authorization errors.

  Note: The MVS programmer sets these options. For detailed information, see "Choosing the RACF access control module customization options" on page 8.
- The classes that the module is trying to use
- The classes for which a RACROUTE request was successful
- Whether the module fully supports Db2 roles

These messages are routed only to the system log and occur only at Db2 initialization time, not during authorization checking. Therefore, these messages are issued regardless of whether any authorization checks have been made, and are issued even when Db2 initialization fails.

Chapter 4. Defining classes for the RACF access control module

You can define classes for RACF access control module if you choose not to use the default classes.

Defining classes for the RACF access control module is optional.

When you change the &CLASSOPT or &CLASSNMT assembler SET symbols from their default values, you must define your own classes in the RACF class descriptor table (CDT).

Tip: If you define your classes in the dynamic class descriptor table instead of the static class descriptor, you do not need to re-IPL to activate the new classes.

It is not necessary to define classes for Db2 objects and administrative authorities that are not protected by the RACF access control module.

You can define classes for Db2 objects and you can define classes for administrative authorities.

When using the single-subsystem scope, the RACF access control module builds class names dynamically by concatenating the Db2 subsystem name, or group attachment name, with the object type. As a result, multiple Db2 subsystems can use the same copy of the RACF access control module. However, you must create an installation-defined set of classes for each subsystem.

When using the multiple-subsystem scope, the RACF access control module builds class names dynamically by concatenating the &CLASSNMT with the object type. As a result, any Db2 subsystem with the same &CLASSNMT can use the same copy of the RACF access control module. You can create an installation-defined set of classes for each subsystem or you can choose to use the supplied classes instead.

Restrictions:

1. If you choose to use installation-defined classes, you must use installation-defined classes with all objects for the same copy of the RACF access control module. You cannot mix classes supplied by IBM and installation-defined classes. To use both types, you must use different versions of the RACF access control module.
2. RACF expects that installation-defined classes have the same class descriptor table attributes as the corresponding Db2 classes supplied by IBM.

Related concepts

Supplied RACF resource classes for Db2

Defining class names for Db2 objects

Defining class names for administrative authorities

RACF security administrators can create profiles with specific Db2 administrative authorities that allow users to access resources.

Related reference

RACF security administration (Security Server RACF Security Administrator's Guide)

Defining class names for Db2 objects

In the supplied class descriptor table (ICHRRCDX), two classes are defined for each Db2 object type (except for the Db2 view object, which shares classes with the table object, and the role and trusted context objects, which are not protected by resource classes), so that each object type has an associated member class and an associated grouping class.

Installations defining their own classes can also define two classes for each object type, if you want member and grouping classes. If only one class is defined for each object type, the class name must begin with M (*not* G).

The actual format of the class names of Db2 objects depends on the classification model being used. The default name for the Db2 administrative authorities class is DSNADM. You can define an additional RACF class.

Related concepts

Supplied RACF resource classes for Db2

Db2 object types

Each authorization request has an associated Db2 object type.

Related tasks

Protecting Db2 administrative authorities

The RACF access control module supports the Db2 concept of administrative authorities.

Defining class names for Db2 objects in single-subsystem scope

When you select this model, the RACF access control module inserts the Db2 subsystem name, or group attachment name, when it constructs RACF class names.

The classes that you define must follow this format:

```
ayyyyxxz
```

where:

a
is M for member class or G for grouping class

yyyy
is the Db2 subsystem name or, if data sharing, the Db2 group attachment name (from XAPLGPAT)

xx
is the type of Db2 object

z
is the &CHAROPT value (The default is 1.)

In single-subsystem scope, the class names of the Db2 object classes contain the Db2 subsystem name or Db2 group attachment name but the profile names of resources in those classes do not. Therefore, in single-subsystem scope, you must define a separate class name for each subsystem that uses the RACF access control module.

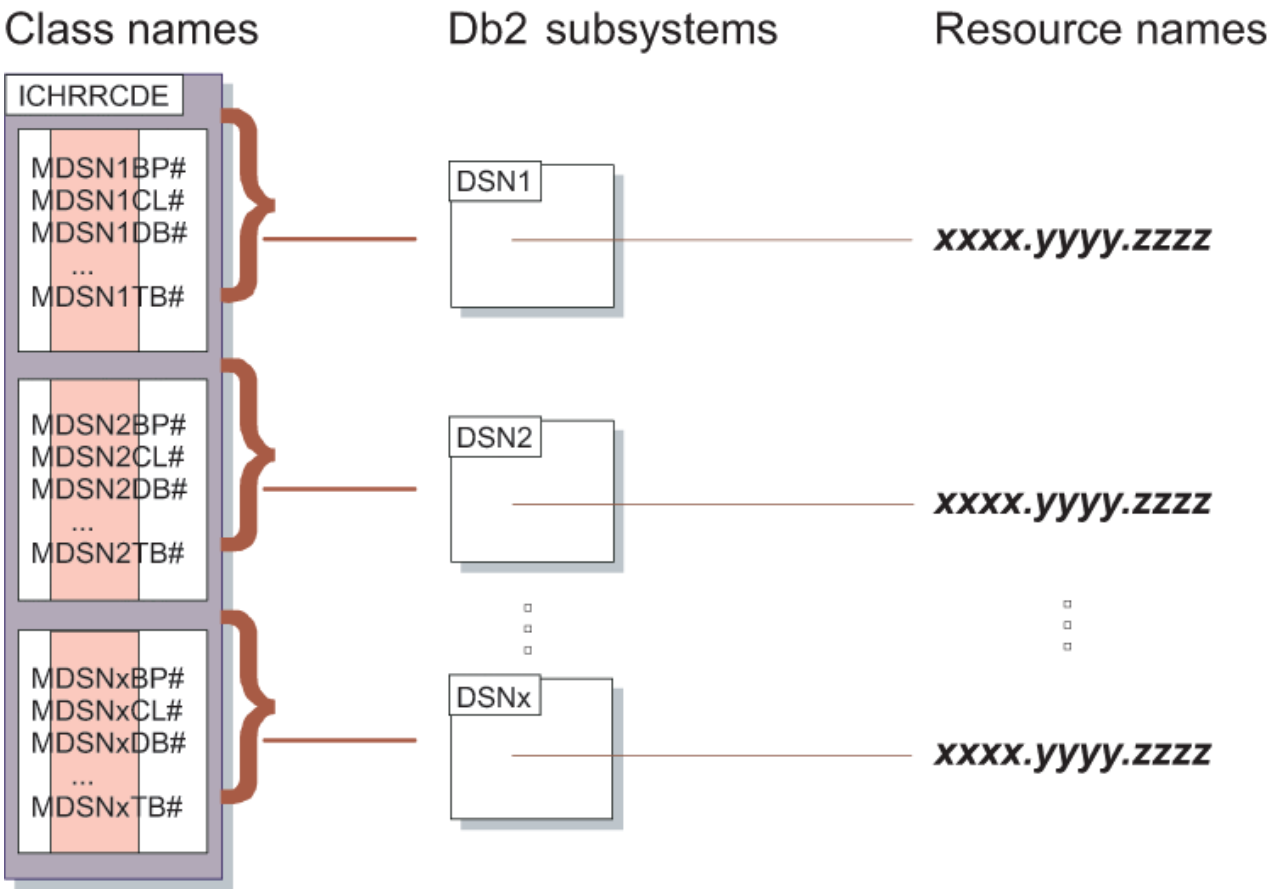

Figure 2. Single-subsystem scope classes

When you use the single-subsystem scope, you cannot use the classes provided in the supplied class descriptor table (ICHRRCDX) unless you are using the default Db2 subsystem name DSN and have altered the &CHAROPT variable in the RACF access control module to be a blank character (''). However, in single-subsystem scope, you must still define a separate class name for every other subsystem that shares the RACF access control module.

When you define your own classes, you can define two classes for each object type if you want both member and grouping classes. If only one class is defined for each object type, the class name must begin with M (*not* G).

Related concepts

Db2 object types
Each authorization request has an associated Db2 object type.

Related reference

RACF security administration (Security Server RACF Security Administrator's Guide)

Defining class names for Db2 objects in multiple-subsystem scope

When you select this model, the RACF access control module places the Db2 subsystem name in the resource name.

Class names that you define must have the following format:

```
abbbbxxz
```

where:

a
Is M for member class or G for grouping class

bbbb
Is the &CLASSNMT value (the default value is DSN)

xx
Is the type of Db2 object (see "Db2 object types" on page 23 for valid values)

z
Is the &CHAROPT value (ignored if &CLASSNMT='DSN')

In multiple-subsystem scope, profile names of resources in the Db2 object classes are prefixed with the Db2 subsystem name, or group attachment name, but the class names are not. See the following figure.

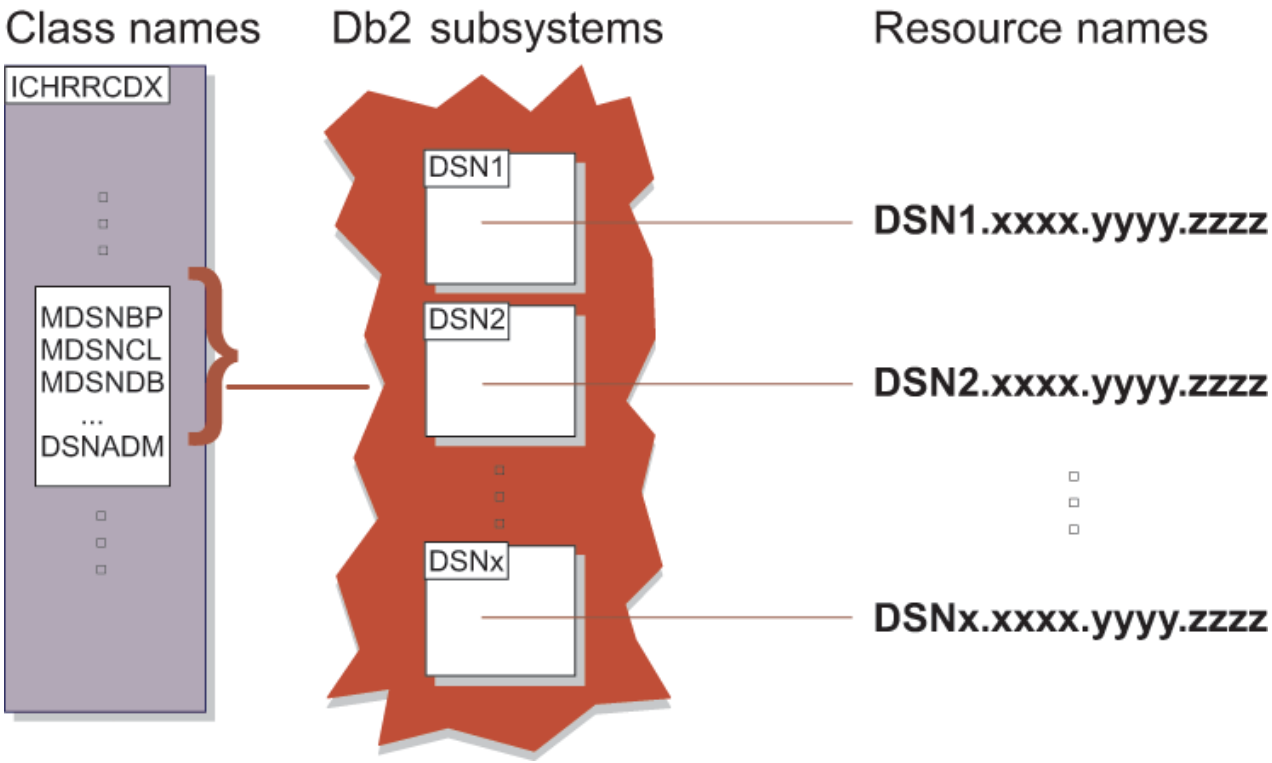

Figure 3. Multiple-subsystem scope classes

If you use the multiple-subsystem scope and the default &CLASSNMT value (DSN), you can use the classes provided in the supplied class descriptor table (ICHRRCDX). Any subsystem sharing the RACF access control module can share the same set of classes. You are not required to define a separate set of classes for each subsystem.

You can change &CLASSNMT if you do not want to use the default (DSN) value. However, if you set &CLASSNMT to a value other than DSN, you must define classes in the class descriptor table (CDT). You can define two classes for each object type if you want both member and grouping classes. If only one class is defined for each object type, the class name must begin with M (*not* G).

Related reference
RACF security administration (Security Server RACF Security Administrator's Guide)

Defining class names for administrative authorities

RACF security administrators can create profiles with specific Db2 administrative authorities that allow users to access resources.

The Db2 administrative authority class (named DSNADM, by default) allows RACF security administrators to create profiles that are suffixed with specific Db2 administrative authorities, to allow users to access

certain resources for specified Db2 subsystems or groups. The format is dependent on the scope (&CLASSOPT) specified.

Defining class names for Db2 administrative authorities in single-subsystem scope

When you select &CLASSOPT 1, the RACF access control module places the Db2 subsystem name, or group attachment name, in the administrative authority class name.

Define administrative authority class names in single-subsystem scope using this format:

```
yyyyADMz
```

where:

yyyy
Is the Db2 subsystem name or, if data sharing, the Db2 group attachment name (from XAPLGPAT)

ADM
Is the designation for administrative authority classes

z
Is the &CHAROPT value (the default value is 1)

In single-subsystem scope, the class names of the Db2 administrative authority classes contain the Db2 subsystem name, or Db2 group attachment name, but the profile names of resources in those classes do not. Therefore, in single-subsystem scope, you must define a separate class name for each subsystem that uses the RACF access control module.

When you select single-subsystem scope, you cannot use the Db2 administrative authority class called DSNADM that is provided in the supplied class descriptor table (ICHRRCDX). You must define your own class in the class descriptor table (CDT), unless you use the default Db2 subsystem name DSN and have altered the &CHAROPT variable in the RACF access control module to be a blank character (' '). However, in single-subsystem scope, you must still define a separate class name for every other subsystem that shares the RACF access control module.

Related reference
RACF security administration (Security Server RACF Security Administrator's Guide)

Defining class names for Db2 administrative authorities in multiple-subsystem scope

You must define administrative authority class names in a specific format when you use the multiple-subsystem scope.

When you select &CLASSOPT 2 or allow it to default, the RACF access control module does not use the Db2 subsystem name or group attachment name in the class name for administrative authorities. Define administrative authority class names in multiple-subsystem scope using this format:

```
yyyyADMz
```

where:

yyyy
Is the &CLASSNMT value (the default value is DSN)

ADM
Is the designation for administrative authority classes

z
Is the &CHAROPT value, which is ignored if &CLASSNMT is set to DSN

In multiple-subsystem scope, profile names of resources in the Db2 administrative authority class are prefixed with the Db2 subsystem name, or Db2 group attachment name, but the class names are not. Therefore, installations using multiple-subsystem scope and the default &CLASSNMT value (DSN) can

use the default Db2 administrative authority class (DSNADM) provided in the supplied class descriptor table (ICHRRCDX). Any subsystem sharing the RACF access control module can share the same class. A separate class does not need to be defined for each Db2 subsystem.

If you set &CLASSNMT to a value other than DSN, you must define a Db2 administrative authority class in the class descriptor table (CDT).

Related reference

RACF security administration (Security Server RACF Security Administrator's Guide)

Chapter 5. Protecting Db2 objects

The resources that apply to a particular invocation of the RACF access control module depend on the input object type and the privilege being checked.

The object types and the names of their associated privileges are shown in Chapter 15, “RACF authorization checking reference,” on page 83. See the Db2 macro DSNXAPRV in *prefix*.SDSNMACS to find the numeric XAPLPRIV values (used by the RACF access control module) that correspond to the privilege names.

The RACF access control module constructs general resource class and profile names for Db2 objects based on the options you specified using the assembler SET symbols:

SET symbol	Default value	Description
&CLASSOPT	2	Specifies the classification model
&CLASSNMT	DSN	Specifies the class name root
&CHAROPT	1	Specifies the class name suffix

The &CLASSOPT, &CLASSNMT, and &CHAROPT options specify the format of the class names and resource profile names used by the RACF access control module. These options are global for each Db2 subsystem, and must be the same for all classes. Each instance of the RACF access control module can only be set up to process one classification model or the other, but not both. See “Choosing the RACF access control module customization options” on page 8 for more information.

If your installation is using the default values for these options, you can use the classes in the supplied class descriptor table (ICHRRCDX). Additional classes do not need to be defined.

Security administrators must define the RACF resources to protect Db2 objects using names that correspond to the format required by the options set in the RACF access control module. The formats for the resource profile names are described in “Defining resource names for Db2 objects” on page 24.

Db2 object types

Each authorization request has an associated Db2 object type.

Db2 provides the object type as a 1-character abbreviation in the XAPLTYPE field. This abbreviation is used by the RACF access control module in conjunction with the code for the requested privilege to determine the authorization check to perform.

A non-valid XAPLTYPE or XAPLPRIV passed to the RACF access control module during authorization checking will cause the RACF access control module to return a return code of 4 ("RACF access not determined; perform Db2 access checking").

the following table lists the Db2 objects, the Db2 abbreviations used in the XAPL, and the abbreviations used in the RACF general resource grouping and member class names (GDSN*xx* and MDSN*xx*):

Table 4. Db2 object abbreviations

Db2 object	Db2 object abbreviation	RACF class abbreviation
Buffer pool	B	BP
Collection	C	CL
Database	D	DB
Java™ archive (JAR)	J	JR
Package	K	PK

Table 4. Db2 object abbreviations (continued)

Db2 object	Db2 object abbreviation	RACF class abbreviation
Plan	P	PN
Role	L	*none*
Schema	M	SC
Sequence	Q	SQ
Storage group	S	SG
Stored procedure	O	SP
System	U	SM
Table or index	T	TB
Table space	R	TS
Trusted context	N	*none*
User-defined distinct type	E	UT
User-defined function	F	UF
View	V	TB

Defining resource names for Db2 objects

The RACF access control module builds resource names depending on the classification model being used.

For single-subsystem scope, the general format for resource name is:

```
[object-name.]privilege-name
```

For multiple-subsystem scope, the general format for resource name is:

```
Db2-subsystem.[object-name.]privilege-name
or, if data sharing:
Db2-group-attachment-name.[object-name.]privilege-name
```

For multiple-subsystem scope, the RACF access control module obtains the Db2 subsystem name, or group attachment name, from XAPLGPAT.

The RACF access control module uses resource names that are based on the object names and the associated privilege names. See “Db2 object types and object names” on page 25 and “Privilege names” on page 26.

Using generic RACF profiles

You can define a RACF resource that protects one or more Db2 objects that have the same security requirements by using generic RACF profiles.

Using generic profiles allows you to exploit naming conventions and greatly reduce the number of RACF profiles you must define. Most generic profiles contain one or more masking characters to replace one or more characters or qualifiers of a resource name.

Related reference

RACF security administration (Security Server RACF Security Administrator's Guide)

Db2 object types and object names

The RACF access control module constructs the RACF resources name using information passed in various fields (XAPLOBJN, XAPLOWNR, and XAPLREL2).

The content of these fields depends on the input object type, XAPLTYPE.

The following table defines the object name qualifiers used in RACF resource names for each Db2 object type:

Table 5. Db2 object name qualifiers for RACF resources

Db2 object	Object name qualifiers
buffer pool	*bufferpool-name*
collection	*collection-ID*
database	*database-name*
Java archive (JAR)	*schema-name.JAR-name*
package	*collection-ID.package-ID* *collection-ID* *owner*
plan	*plan-name* *owner*
role	not applicable
schema	*schema-name* *schema-name.function-name* *schema-name.procedure-name* *schema-name.type-name*
sequence	*schema-name.sequence-name*
storage group	*storage-groupname*
stored procedure	*schema-name.procedure-name*
system	*owner* (BINDAGENT only)
table, index	*table-qualifier.table-name* *table-qualifier.table-name.column-name*
table space	*database-name.table-space-name*
trusted context	not applicable
user-defined distinct type	*schema-name.type-name*
user-defined function	*schema-name.function-name*
view	*view-qualifier.view-name* *table-qualifier.table-name* *table-qualifier.table-name.view-qualifier.view-name*

Note: The format of the Db2 object name qualifiers is defined by Db2.

Long object names

Some Db2 objects can have names containing up to 128 characters.

Because RACF profile names are limited to 246 characters, the RACF access control module might truncate portions of the profile names when you use long object names.

The schema name or table qualifier portion of the profile name might be truncated to 100 characters. For example, consider the RACF profile name for the USAGE privilege on a JAR object:

```
db2-subsystem.schema-name.JAR-name.USAGE
```

The schema name and JAR name can each contain a maximum of 128 characters. If the Db2 subsystem name is four characters, the length of the profile name would reach 268 characters and exceed the maximum name length unless the RACF access control module truncates the schema name to 100 characters.

Variables for INSERT, UPDATE, and DELETE operations on views are also truncated to specific lengths. The table-qualifier and the view-qualifier are truncated at 32 characters each, and the table-name and the view-name are truncated at 64 characters each. For example, consider the RACF profile name for the INSERT privilege on a view.

```
db2-subsystem.table-qualifier.table-name.view-qualifier.view-name.INSERT
```

```
db2-subsystem.table-qualifier.table-name.column-name.UPDATE
```

When you use long object names, truncation can cause unintended results when you also use discrete RACF profiles. If truncation occurs, a single discrete profile might inadvertently protect multiple similarly named resources when the first 100 characters of the schema names are the identical and the qualified object names, such as JAR name, subsystem name, and privilege name, are also identical.

Privilege names

The RACF access control module constructs the Db2 resource name using the Db2 privilege name as the lowest-level qualifier (RACF profile-name suffix) in the resource name.

Each explicit privilege used as a low-level qualifier corresponds to one of the explicit privilege names that Db2 uses for a particular object. For a complete reference of all valid privilege names that can be used in a resource name for each Db2 object, see the tables in Chapter 15, “RACF authorization checking reference,” on page 83.

Tip: You can authorize a user for one or more privileges on a Db2 object by defining a generic RACF profile using an asterisk (*****) in place of the privilege name and then permitting the user to the generic profile. However, if a more specific generic profile or a discrete profile also protect the same privilege or set of privileges, RACF will use those profiles to control access rather than the less specific generic profile.

See “Db2 GRANT statements” on page 50 for an example of using a generic character in place of the privilege name. (In contrast with SQL, in RACF a single asterisk (*****) matches characters within the scope of a single qualifier.)

Chapter 6. Protecting Db2 administrative authorities

The RACF access control module supports the Db2 concept of administrative authorities.

About this task

Db2 administrative authorities often include privileges that are not explicit, have no name, and cannot be specifically granted. For example, the ability to terminate any utility job is included in the SYSOPR authority.

During authorization checking, if a user is not permitted access to the object through the object's resource profile, subsequent checks are made to determine if the user has been permitted access to system resources through their administrative authorities. These checks are made using profiles in the Db2 administrative authority class DSNADM. Db2 includes the SQLADM administrative authority in the MDSNSM GDSNSM classes.

The administrative authorities that apply to a particular invocation of the RACF access control module, depend on the input object type (XAPLTYPE) and the privilege being checked (XAPLPRIV).

Like the names used to protect Db2 objects, the general resource class and profile names used to protect Db2 administrative authorities depend on the options specified with the assembler SET symbols.

Related reference

Administrative authorities (Managing Security)

RACF authorization checking reference

You can use the RACF access control module to perform RACF authorization checking for several Db2 objects.

Defining resource names for administrative authorities

The RACF access control module builds the resource names for administrative authorities based on the classification model you selected.

About this task

For single-subsystem scope, the format for Db2 administrative authority resources is:

```
[object-name.]authority-name
```

For multiple-subsystem scope, the general format is:

```
Db2-subsystem.[object-name.]authority-name
```

or, if data sharing,

```
Db2-group-attachment-name.[object-name.]authority-name
```

For multiple-subsystem scope, the Db2 subsystem name or Db2 group attachment name is obtained from XAPLGPAT. The object name used depends on the Db2 administrative authority. See “Db2 administrative authorities and object names” on page 27.

Db2 administrative authorities and object names

The RACF access control module constructs the RACF resource name using information that is passed in XAPLOBJN, XAPLOWNQ, or XAPLREL2.

The content of these fields depends on the input object type, XAPLTYPE.

These checks are made using profiles in the Db2 administrative authority class DSNADM. Db2 also includes the SQLADM administrative authority in the systems class MDSNSM GDSNSM.

This table lists the Db2 administrative authorities and the associated RACF object qualifiers:

Table 6. Db2 administrative authorities and object qualifiers

Administrative authority	RACF object qualifier
ACCESSCTRL	—
DATAACCESS	—
DBADM	*database-name*
DBCTRL	*database-name*
DBMAINT	*database-name*
PACKADM	*collection-ID*
SECADM	—
SQLADM	—
SYSADM	—
SYSCTRL	—
SYSDBADM	—
SYSOPR	—

Note: The format of the Db2 object names is defined by Db2.

Related reference
TABLE_NAME (Db2 SQL)

Chapter 7. Making your new RACF resources effective

You must take several steps to ensure that your new resource definitions are effective.

About this task

If your Db2 subsystem was up and running when you defined your new Db2 objects and administrative authorities in Chapter 5, “Protecting Db2 objects,” on page 23 and Chapter 6, “Protecting Db2 administrative authorities,” on page 27, your new resource definitions are not in effect until you take explicit steps to make them effective. In order to be effective, the new RACF resource definitions must be read into storage for RACF access list checking.

Depending on whether the resource classes where you defined the new resources were active at the time your Db2 subsystem was started, you execute different sets of commands to put your resource definitions in effect, as shown below.

If the class was not active

When you define new RACF resources to protect Db2 objects, you must ensure that the new resource definitions become effective.

About this task

In a class that was not active at Db2 startup time, you must stop the Db2 subsystem, activate the class, and then restart the Db2 subsystem. Restarting the Db2 subsystem reads the new profiles into storage and allows the new resource definitions to become effective.

Example

From the MVS console, issue the following command:

```
-STOP DB2
```

Issue the following RACF commands:

```
SETROPTS CLASSACT(classname)
```

From the MVS console, issue the following command:

```
-START DB2
```

If the class was active

When the class was active at Db2 startup time, you can dynamically refresh all the profiles in storage for this class and allow the new resource definitions to become effective by issuing the following RACF command.

About this task

You do not need to restart the Db2 subsystem after you execute the RACLIST command.

Example

Issue the following RACF command:

```
SETROPTS RACLIST(classname) REFRESH
```

Chapter 8. Debugging the RACF access control module

You can use IFCID 0314 trace records to obtain the parameter list on return from the RACF access control module.

To generate IFCID 0314 records, start performance trace class 22.

You can generate IFCID 0314 trace records to identify authorization checks that use Db2 security facilities. Doing this can aid you in converting from using Db2 security facilities to using the RACF access control module for security. To generate IFCID 0314 records only for authorization checks that use Db2 security facilities, in addition to starting a performance class 22 trace, start a trace for IFCID 0410.

You can correlate IFCID 0314 records and RACF SMF records by timestamp to determine which SMF record is associated with each IFCID record.

Related concepts

Performance trace (Db2 Performance)

Dump titles for the RACF access control module

The RACF access control module generates dump titles.

The RACF access control module generates the following dump titles:

```
COMPON=DB2,COMPID=5740DRE00,ISSUER=DSNX@FRR,MODULE=DSNX@XAC,
ABEND=S0sss,REASON=NONE    ,L=zzzzzzzz

COMPON=DB2,COMPID=5740DRE00,ISSUER=DSNX@FRR,MODULE=DSNX@XAC,
ABEND=S0sss,REASON=aaaaaaaa,L=zzzzzzzz

COMPON=DB2,COMPID=5740DRE00,ISSUER=DSNX@FRR,MODULE=DSNX@XAC,
ABEND=Uuuuu,REASON=NONE    ,L=zzzzzzzz

COMPON=DB2,COMPID=5740DRE00,ISSUER=DSNX@FRR,MODULE=DSNX@XAC,
ABEND=Uuuuu,REASON=aaaaaaaa,L=zzzzzzzz
```

where:

sss
: is the system abend code

uuuu
: is the user abend code

aaaaaaaa
: is the abend reason code

zzzzzzzz
: is the module length

Using the content of XAPLDIAG

The RACF access control module returns a parameter, XAPLDIAG, that Db2 and other licensed programs can use to trap and obtain diagnostic information.

When the RACF access control module issues the RACROUTE REQUEST=FASTAUTH macro for authorization checking, depending on the AUDIT options used with the check, the module can record the resulting SAF return code, RACF return code, and RACF reason code in XAPLDIAG. Each invocation of the RACF access control module can issue multiple RACROUTE REQUEST=FASTAUTH macros, but the module evaluates each return code generated and determines the single correct return code to send to Db2.

The RACF access control module can store up to 20 sets of return codes from RACROUTE REQUEST=FASTAUTH macros in XAPLDIAG, allowing the results of a specific RACROUTE REQUEST=FASTAUTH macro to be determined.

The XAPL parameter list can be captured using Db2 trace record IFCID 314. In addition, the return code and corresponding reason code (EXPLRC1 and EXPLRC2) for authorization failures are captured in Db2 trace record IFCID 140.

The content of XAPLDIAG depends on the return code and reason code from the RACF access control module. The return and reason codes in XAPLDIAG are in the same order as the checks that are described in the rules table for each privilege. You can use this order to determine which checks failed and which checks granted access.

- If EXPLRC1=4 and ECPLRC2=14 (decimal), the ALESERV failed and the module made no RACROUTE REQUEST=FASTAUTH checks. In this case the first word of XAPLDIAG contains the non-zero ALESERV return code.
- Otherwise, each word of XAPLDIAG can contain a SAF return code, RACF return code, and RACF reason code corresponding to a non-zero return code from a RACROUTE REQUEST=FASTAUTH macro. Information related to non-zero return codes is stored in XAPLDIAG beginning with the first word until information related to all non-zero return codes has been stored, or until the XAPLDIAG area has filled. XAPLDIAG contains 20 words, allowing information related to 20 FASTAUTH requests to be stored for an invocation of the RACF access control module. If more than 20 FASTAUTH requests are issued, only the first 20 sets of return codes are stored.

DBADM authorization checking for the CREATE VIEW privilege can result in more than 20 FASTAUTH requests because a CREATE VIEW request can reference tables, or a combination of tables and views, from multiple databases. Db2 passes the names of all the databases referenced in the CREATE VIEW using a database list pointed to by XAPLDBSP. If SYSCTRL or SYSADM authorization checking does not grant the CREATE VIEW privilege and the XAPLCRVW field indicates that DBACRVW is enabled, the RACF access control module checks the user's DBADM authorization for each database in the list. The result of each DBADM check is placed in the XAPLDBDA field associated with each database. The RACF access control module updates XAPLDBDA with the following codes:

Y
Access to the database is allowed.

N
Access to the database is not allowed.

U
RACF was unable to return a decision. This occurs when the FASTAUTH request returns a SAF return code of X'04'.

The database list pointed to by XAPLDBSP is made up of four-word database information structures mapped by the XAPLDBS macro.

```
XAPLDBNP DS F     PTR TO NEXT DATABASE INFORMATION STRUC
XAPLDBNM DS CL8   DATABASE NAME
XAPLDBDA DS CL1   'Y' - IS DBADM
XAPLDBIM DS CL1   'Y' - IS AN IMPLICIT DATABASE
XAPLRSV5 DS CL2   RESERVED - UNUSED
```

Although DBADM checks can be done for multiple databases, only the results of the first 20 FASTAUTH requests are stored in XAPLDIAG. The results of all DBADM checking for each database is contained in the XAPL parameter list and is available using Db2 trace record IFCID 314.

The RACF access control module truncates the SAF return codes and RACF return codes to one byte, and the RACF reason code to two bytes, before storing them in XAPLDIAG. The format of each word in XAPLDIAG is:

```
xxyyzzzz
```

where:

xx
is the 1-byte SAF return code

yy
is the 1-byte RACF return code

zzzz
is the 2-byte RACF reason code

Related concepts

“Common problems and considerations” on page 56
If you define special classes in the class descriptor table, you might encounter some common problems.

Related reference

Authorization checking (XAPLFUNC = 2)
The RACF access control module requires an input ACEE to perform authority checking.

z/OS Security Server RACROUTE Macro Reference

Parameter list for the access control authorization routine

An authorization routine's parameter list points to other information.

The following figure shows how the parameter list points to other information.

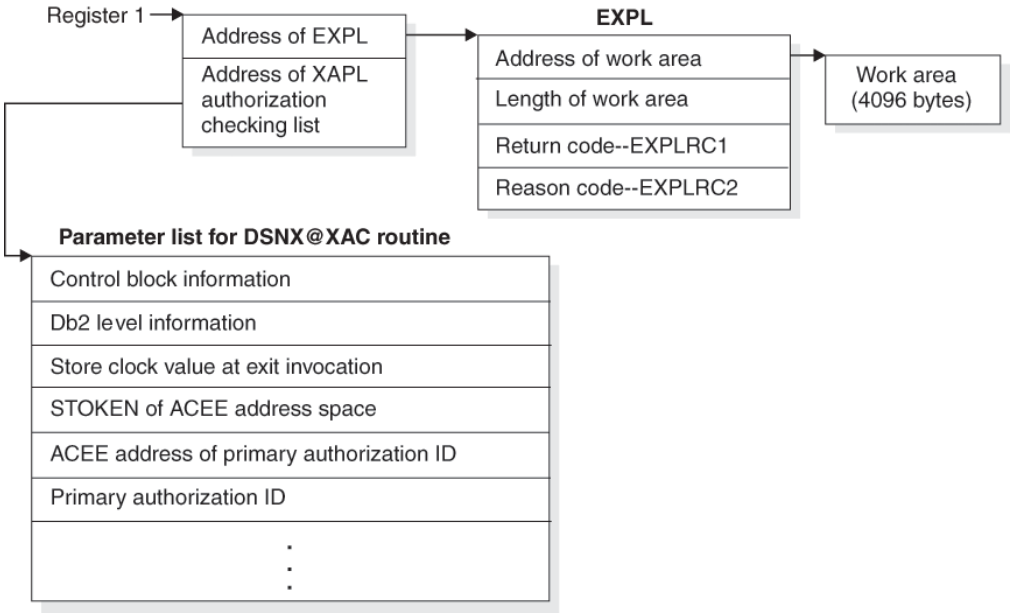

Figure 4. How an authorization routine's parameter list points to other information

The work area (4096 bytes) is obtained once during the startup of Db2 and only released when Db2 is shut down. The work area is shared by all invocations of the RACF access control module.

Related reference

Parameter list for access control authorization routines (Managing Security)

Implicit privileges of ownership

The RACF access control module performs the checks for implicit privileges of ownership.

For an implicitly created database, the module must also check the ownership of other objects, such as the table space or index space. The owner of the other object in the decision is in the XAPLOOON and XAPLOOOT fields. The other object is in the XAPLOONM field. The following table shows these checks.

Table 7. Checks for implicit privileges of ownership

Type of owner (XAPLOWRT)	Type of authorization ID checked (XAPLUCKT)	Checks performed	Reason code (EXPLRC2)

Authorization ID	Authorization ID	If XAPLOWAC is on, XAPLUPRM is set to the ID that Db2 performs authorization checking (XAPLUCHK)
		`XAPLOWNR=XAPLUCHK`
		`XAPLOWNR=XAPLUPRM`
		If XAPLACAC is on, RACF does not perform the check for XAPLOWNR=XAPLUCHK.
			13

Authorization ID	Role	`XAPLOWNR=XAPLUPRM`	13
Role	Authorization ID	`XAPLOWNR=XAPLROLE`	16

Role	Role	If XAPLFLG1=B'1xxxxxxx':
		`XAPLOWNR=XAPLUCHK`
		`XAPLOWNR=XAPLROLE`
		If XAPLFLG1=B'0xxxxxxx':
		`XAPLOWNR=XAPLUCHK`
			16

Table 8. Checks for implicit privileges of ownership of table and index spaces in implicitly created databases

Type of owner (XAPLOOOT)	Type of authorization ID checked (XAPLUCKT)	Checks performed	Reason code (EXPLRC2)

Authorization ID	Authorization ID	If XAPLOWAC is on, XAPLUPRM is set to the ID that Db2 performs authorization checking (XAPLUCHK)
		`XAPLOOON=XAPLUCHK`
		`XAPLOOON=XAPLUPRM`
		If XAPLACAC is on, RACF does not perform the check for XAPLOOON=XAPLCHK.
			17

Authorization ID	Role	`XAPLOOON=XAPLUPRM`	17
Role	Authorization ID	`XAPLOOON=XAPLROLE`	18

Table 8. Checks for implicit privileges of ownership of table and index spaces in implicitly created databases (continued)

Type of owner (XAPLOOOT)	Type of authorization ID checked (XAPLUCKT)	Checks performed	Reason code (EXPLRC2)

Role	Role	If XAPLFLG1=B'1xxxxxxx':
		`XAPLOOON=XAPLUCHK`
		`XAPLOOON=XAPLROLE`
		If XAPLFLG1=B'0xxxxxxx':
		`XAPLOOON=XAPLUCHK`
			18

Authorization and ownership checking with roles

You can use the RACF access control module to perform ownership checking with roles.

The tables below show the ownership and authorization checks that the RACF access control module performs. The ownership checks are performed first, then the authorization checks. You can use these tables with trace data to diagnose problems.

The following table expands on the information in “Implicit privileges of ownership” on page 33.

Table 9. Ownership checks with roles

XAPLONRT (type of ID that owns object)	XAPLOWNR (owner of object)	XAPLCHKS (bit 8 in XAPLFLG1)	XAPLUCKT (type of ID being checked by Db2)	XAPLUCHK (authorization ID or role being checked by Db2)	XAPLROLE (role associated with requester)	XAPLUPRM (requester - always an authorization ID)	Action
Blank (indicates authorization ID)	Authorization ID	Not applicable	Blank (indicates authorization ID	Authorization ID	Role	User ID	Does XAPLOWNR = XAPLUPRM? Does XAPLOWNR = XAPLUCHK? If either matches, the ownership check passes.RACF does not check for XAPLOWNR = XAPLUCHK if XAPLACAC='1'B and XAPLONRT is a blank and XAPLUCKT is a blank.
Blank (indicates authorization ID)	Authorization ID	Not applicable	"L" (indicates a role)	Role	Role	User ID	Compare XAPLOWNR to XAPLUPRM. If equal, the ownership check passes.
"L" (indicates a role)	Role	Not applicable	Blank (indicates authorization ID	Authorization ID	None	User ID	The ownership check fails because the owner is a role and nothing else is a role.
"L" (indicates a role)	Role	Bit = "ON"	Blank (indicates authorization ID	Authorization ID	Role	User ID	Compare XAPLOWNR to XAPLROLE. If equal, the ownership check passes.
"L" (indicates a role)	Role	Bit = "ON"	"L" (indicates a role)	Role	Role	User ID	Does XAPLOWNR = XAPLROLE? Does XAPLOWNR = XAPLUCHK? If either matches the ownership check passes.

Table 9. Ownership checks with roles (continued)

XAPLONRT (type of ID that owns object)	XAPLOWNR (owner of object)	XAPLCHKS (bit 8 in XAPLFLG1)	XAPLUCKT (type of ID being checked by Db2)	XAPLUCHK (authorization ID or role being checked by Db2)	XAPLROLE (role associated with requester)	XAPLUPRM (requester - always an authorization ID)	Action
"L" (indicates a role)	Role	Bit = "OFF"	"L" (indicates a role)	Role	Role	User ID	Does XAPLOWNR = XAPLUCHK? If equal the ownership check passes.

Table 10. Authorization checks with roles

Type of privilege	XAPLUCKT (type of ID being checked by Db2)	XAPLCHKS (bit 8 in XAPLFLG1	XAPLROLE (role associated with requester)	XAPLUCHK (authorization ID or role being checked by Db2)	ACEE (requester - always an authorization ID = to XAPLUPRM)	Action
All	Blank (indicates authorization ID)	Not applicable	Blank	Ignored	Authorization ID	Perform FASTAUTH check with AUTHCHKS=ALL
All	Blank (indicates authorization ID)	Not applicable	Role	Ignored	Authorization ID	Perform FASTAUTH check with AUTHCHKS=ALL. The check includes the role, from XAPLROLE.
All except those that occur during a create or bind	"L" (indicates a role)	Bit = "ON"	Role	Ignored	Authorization ID	Perform FASTAUTH check with AUTHCHKS=ALL. The check includes the role, from XAPLROLE.
All that occur during a create or bind	"L" (indicates a role)	Bit = "OFF"	Role (ignored)	Role	Authorization ID	Perform FASTAUTH check with AUTHCHKS=CRITONLY. Check only the role, from XAPLUCHK.

Note: XAPLUCHK can contain a role.

Chapter 9. Auditing for the RACF access control module

The RACF access control module provides RACF resource profiles to check authorization for Db2 privileges and authorities.

RACF resource profiles represent the various Db2 privileges. You can use the RACF auditing tools to extract the information that you need.

You can use the SMF data unload utility or the RACF report writer to extract and format the SMF records. When the RACF access control module uses a RACROUTE REQUEST=FASTAUTH request to create an audit record, the record contains log string data that includes additional diagnosis information described in "Using log string data" on page 38. You can use the log string information to link Db2 trace record IFCID 314 and a corresponding RACF SMF record.

In addition, you can use the RACF informational messages. For more information, see "RACF informational messages" on page 15.

Example of resource checking

RACF resources are checked when a user issues the SELECT statement.

The following example shows the series of RACF resources that are checked when a user issues the SELECT statement.

When RACF checks authorization, the requester must own the object or have at least READ access to one of the following profiles:

Profile name	Class	Note
subsystem.table-qualifier.table-name.SELECT	MDSNTB	Gives access to the table
subsystem.database-name.DBADM	DSNADM	Gives access to the database that holds the table
subsystem.SYSCTRL	DSNADM	Bypassed for user tables
subsystem.SYSADM	DSNADM	—

RACF produces an SMF record for a failure only after checking the entire list of profiles and the requester fails to meet any of the requirements. RACF does not produce an audit record if:

- The requester meets any of the requirements and access is granted, or
- The RACF access control module returns the authority checking responsibility to Db2.

If Db2 objects are defined to RACF using the WARNING option, you receive ICH408I messages that identify those profiles that would fail a request and the requested access is allowed.

Note: For Db2 releases before Db2 V8, the ICH408I messages were suppressed.

If the WARN option is added to a resource that is requested by a user with a Db2 administrative authority, such as SYSADM, DBADM or in some cases, SYSCTRL, that normally allows the user to access the object, the user can ignore the WARNING message.

An audit record is produced for the first resource that has auditing indicated by the covering profile and receives a return code of 8.

RACF produces an SMF record for a success when the requester indicates that it must be performed.

For a list of the RACF classes, see Chapter 13, “Supplied RACF resource classes for Db2,” on page 73. For a full list of each RACF resource checked for each privilege, see Chapter 15, “RACF authorization checking reference,” on page 83.

Using log string data

The log string data contains information that can help you audit Db2 successfully.

Db2 uses the XAPL parameter list (DSNDXAPL macro) to pass log string information to the RACF access control module. The LOGSTR= parameter of the RACROUTE REQUEST=FASTAUTH request contains the input portion of XAPL and does the following:

- Identifies the RACF access control module request that caused RACF to create the audit record. The RACF profile causing the audit record to be cut could be a profile that provides a Db2 administrative authority and might not identify the specific Db2 resource being accessed. The log string data contains values from the XAPL parameter list that are necessary to identify that unique request from the RACF access control module.
- Links SMF type 80 records with Db2 IFCID 314 records. Each invocation of the RACF access control module might produce an SMF type 80 record. Db2 might produce a Db2 IFCID 314 record in addition to the SMF type 80 records cut by RACF. You can determine that the records were cut for the same RACF access control module request if the LOGSTR_TIME and LOGSTR_USER values in the SMF type 80 record match the XAPLSTCK and XAPLUPRM values in the IFCID 314 request. The RACF access control module uses these time and user values created from the log string data to link the RACF and Db2 information.

The following table shows the ordered information included in log string data. A blank space separates each field, as indicated in the table.

Table 11. Information contained in log string data

Log string data	Length	XAPL field name	Description
LOGSTR_DATA	DS 0CL241		
LOGSTR_TIME	DS CL8	XAPLSTCK	Time
	DS CL1		
LOGSTR_USER	DS CL8	XAPLUPRM	User
	DS CL1		
LOGSTR_SUBSYSTEM	DS CL4	XAPLGPAT	Subsystem name, or if data sharing, Db2 group attachment name
	DS CL1		
LOGSTR_OBJTYPE	DS CL1	XAPLTYPE	Object type
	DS CL1		

Table 11. Information contained in log string data (continued)

Log string data	Length	XAPL field name	Description
LOGSTR_FLAGS	DS 0CL16	XAPLFLG1	**Flags:** The flags in this field are declared as BL1. The field is translated to CL16 in the LOGSTR data field and contains one character for each bit with a blank character between each one. • If the bit is on, Y is inserted. • If the bit is off, N is inserted. • Reserved bits are left blank.
LOGSTR_SECNDRY_ID	DS CL1		Secondary ID (Y or N)
	DS CL1		
LOGSTR_USERTAB	DS CL1		User table (Y or N)
	DS CL1		
LOGSTR_AUTOBIND	DS CL1		Autobind authority check (Y or N)
	DS CL1		
LOGSTR_DBCRTVW	DS CL1		DBADM authority to create views for others (Y or N)
	DS CL1		
LOGSTR_RDRW	DS CL1		Read/write request (Y or N)
	DS CL1		
LOGSTR_NOAUDIT	DS CL1		Suppress failure records (Y or N)
	DS CL5		
LOGSTR_OBJNAME	DS CL20	XAPLOBJN	Object name: This is the first 20 bytes of the XAPLOBJN field.
	DS CL1		
LOGSTR_OBJOWNER	DS CL20	XAPLOWNQ	Object owner or qualifier: This is the first 20 bytes of the XAPLOWNQ field.
	DS CL1		
LOGSTR_REL1	DS CL20	XAPLREL1	Related information 1: This is the first 20 bytes of the XAPLREL1 field.
	DS CL1		
LOGSTR_REL2	DS CL20	XAPLREL2	Related information 2: This is the first 20 bytes of the XAPLREL2 field.
	DS CL1		
LOGSTR_PRIV	DS CL3	XAPLPRIV	Privilege
	DS CL1		
LOGSTR_SOURCE	DS CL1	XAPLRSV3	Reserved
	DS CL1		
LOGSTR_CLASS	DS CL8		Class name

Table 11. Information contained in log string data (continued)

Log string data	Length	XAPL field name	Description
	DS CL1		
LOGSTR_ENTY	DS CL100		Entity name: This is the first resource checked for a specific request.

Examples for setting audit controls for Db2

The RACF access control module attempts to produce an audit record after checking the list of profiles.

Example 1

In this example, user ROGERM wants to use the SQL SELECT statement to retrieve information from table ICH in database DSNDB04 on the Db2 subsystem named DSN. The table qualifier is LOVES. (Refer to Chapter 15, “RACF authorization checking reference,” on page 83 for the summary of table checking for the privilege.)

- Does ROGERM own the table?

  Because ROGERM does not own the table, the table name qualifier passed from Db2 does not match the user ID. In this case, RACF does not check a profile, so no audit record is written.
- Does ROGERM have SELECT authority?

  RACF checks DSN.LOVES.ICH.SELECT in classes MDSNTB and GDSNTB. ROGERM does not have the required SELECT authority. If ROGERM doesn't meet any of the other requirements, this is the "first failing resource."
- Does ROGERM have database administrator authority?

  RACF checks DSN.DSNDB04.DBADM in class DSNADM. ROGERM does not have this authority.
- Does ROGERM have system administrator authority?

  RACF checks DSN.SYSADM in class DSNADM. ROGERM does not have this authority.

Because ROGERM has none of the required authorities, RACF produces SMF records relating to the first failure it encountered. Although ROGERM didn't own the table, no profiles were checked and failures were not audited. Therefore, the first failing resource is DSN.LOVES.ICH.SELECT. RACF produces an audit record for this resource and identifies it in message DSN408I. The data is contained in the log string information and can be used in a report.

Example 2

In this example, user ROGERM issues a START DATABASE(DSNDB04) request for Db2 subsystem DSN. (Refer to Chapter 15, “RACF authorization checking reference,” on page 83 for the summary of database checking for the privilege.)

- Does ROGERM have STARTDB authority?

  RACF checks DSN.DSNDB04.STARTDB in classes MDSNDB and GDSNDB. ROGERM does not have the required STARTDB authority. If ROGERM doesn't meet any of the other requirements, this is the "first failing resource."
- Does ROGERM have database maintenance authority?

  RACF checks DSN.DSNDB04.DBMAINT in class DSNADM. ROGERM does not have the required DBMAINT authority.
- Does ROGERM have database control authority?

  RACF checks DSN.DSNDB04.DBCTRL in class DSNADM. ROGERM does not have the required DBCTRL authority.

- Does ROGERM have database administrator authority?

  RACF checks DSN.DSNDB04.DBADM in class DSNADM. ROGERM does not have the required DBADM authority.
- Does ROGERM have system control authority?

  RACF checks DSN.SYSCTRL in class DSNADM. ROGERM does not have this authority.
- Does ROGERM have system administrator authority?

  RACF checks DSN.SYSADM in class DSNADM. ROGERM does not have this authority.

Because ROGERM has none of the sufficient authorities, RACF produces SMF records relating to the failure. The failure record is written for resource DSN.DSNDB04.STARTDB, which was the first failing resource. The log string information can help you to determine what ROGERM wanted to do. It includes the object type, object name, and privilege, which you can use in a report.

Chapter 10. Special considerations

In certain instances, the RACF authorization checking done by the RACF access control module is different from the authorization checking done by Db2.

These instances are described in this section, along with other Db2 authorization considerations.

Materialized query tables

When a materialized query table is created, a create view (CRTVUAUTT) authorization check is performed.

The CRTVUAUTT check is used to determine whether the creator of a materialized query table can provide the required SELECT privileges on base tables to the owner of the materialized query table. If the owner of the materialized query table has the required privileges, then the CRTVUAUTT authorization check proves redundant. However, the check is performed before the owner of the materialized query table's privileges are determined. Therefore, if the materialized query table owner holds the necessary privileges and the creator of the materialized query table does not, the CRTVUAUTT check can produce unwanted error messages. To suppress these unwanted error messages, XAPLFSUP is turned on to indicate that the RACF access control module should suppress these messages.

Db2 data sharing

You can use the RACF access control module with Db2 data sharing.

In a data sharing environment, Db2 passes the Db2 group attachment name to the RACF access control module, instead of a Db2 subsystem name. As a result, class names and profile names must be defined with the Db2 group attachment name. When you use the RACF access control module in a data sharing environment, all subsystems in the Db2 data sharing group must share the same RACF database.

Authorization checking for implicitly created databases

RACF access control module checks only for authorization to DSNDB04. It does not check for authorization to the implicitly created database.

On Db2 V8, if you create a table and do not specify a database name, Db2 creates the table in the default database, DSNDB04. With Db2 V9, Db2 creates a database for you with the name DSN*xxxxx*, where *xxxxx* is a zero-padded increasing integer, and creates the table space or table in that database. The value of *xxxxx* wraps to 00001 after the limit for the number of implicitly created databases is reached. As a result, tables created by different users might be placed in the same implicitly created database.

Db2 allows access to an implicitly created database if the user has authorization to *either* DSNDB04 or the implicitly created database. The RACF access control module differs from Db2 in that it checks *only* for authorization to DSNDB04. It does *not* check for authorization to the implicitly created database.

Authorization checking for operations on views

For most operations on views, the RACF access control module checks for authorization to the view.

For most operations on views, the RACF access control module checks for authorization on the view. Authorization checking for INSERT, DELETE, and UPDATE are different because the operations on views can affect the base tables for the views.

In general, three types of views can be defined:

Updatable view
A view that is defined with simple column references in the SELECT list of the view definition, and a single table in the FROM clause of the view definition. An INSERT, DELETE, or UPDATE operation to the view is reflected to the underlying table.

Read-only view
A view created from multiple tables. The INSERT, DELETE, and UPDATE operations fail for these views.

INSTEAD OF trigger view
The view is read-only, but the SQL in the trigger package can update the underlying table or tables.

For INSERT, DELETE, and UPDATE operations on updatable views, the RACF access control module checks for authorization to the resource name which includes both the underlying table information (qualifier and name) and view information (qualifier and name) and not to the view itself.

For INSERT, DELETE, and UPDATE operations on read-only and INSTEAD OF trigger views, the RACF access control module checks for authorization on the view.

If a view is created on another view, during view creation the RACF access control module does authorization checks for INSERT, DELETE, and UPDATE. These checks are done on the base view.

For more information, see "View privileges" on page 131.

Access to privileges based on factors other than RACF profiles

The security administrator can grant access to privileges by using the RACF profiles.

Several other factors can grant access to privileges. These factors are checked before checking the applicable RACF profiles and they include the following items:

- Implicit privileges of ownership
- Matching schema names
- Ownership of other objects

Implicit privileges of ownership

When a user is the owner of a Db2 object, that user might have some implicit privileges, but not all privileges associated with the object.

The RACF access control module supports certain implicit privileges of ownership for the following Db2 objects and associated privileges.

Table 12. Db2 objects and implicit privileges associated with ownership. The owner of the object is identified by the XAPLOWNR and XAPLONRT fields.

Db2 object	Implicit privileges
Database	DISPLAYDB, MERGECOPY, IMAGCOPY, MODIFY RECOVERY, QUIESCE, RECOVERDB, REPORT, REORG, REPAIR, RUN REPAIR UTILITY, RUN CHECK INDEX/LOB UTILITY, STATS, STARTDB, STOPDB, TERM UTILITY ON DATABASE
Java archive (JAR)	USAGE
Package	BINDAUT, COMMENT ON, COPYAUT
Plan	BINDAUT, COMMENT ON
Role	COMMENT ON, DROP
Sequence	ALTER, COMMENT ON, USAGE
Stored procedure	DISPLAY, EXECUTE, START, STOP
Table	All privileges except CRTSYAUT, DRPSYAUT, CRTVUAUT
Trusted context	COMMENT ON, DROP

Table 12. Db2 objects and implicit privileges associated with ownership. The owner of the object is identified by the XAPLOWNR and XAPLONRT fields. *(continued)*

Db2 object	Implicit privileges
User-defined distinct type	USAGE
User-defined function	DISPLAY, EXECUTE, START, STOP
View	ALTER, COMMENT ON, DROP

To check authorization for the privileges associated with implicit ownership, the RACF access control module uses ownership information passed from Db2 in the XAPLOWNR field of DSNDXAPL.

If the object is owned by an authorization ID, the RACF access control module authorizes access and returns a return code 0 in EXPLRC1 and reason code 13 in EXPLRC2. If the object is owned by the role in effect for the user, the RACF access control module authorizes access and returns a return code 0 in EXPLRC1 and reason code 16 in EXPLRC2.

If these checks fail, for some privileges the RACF access control module checks whether the current authorization ID (in the field XAPLUCHK) matches the schema name.

Note: On multilevel-secure systems with the RACF SETROPTS MLS option active, the ownership check is not performed.

Matching schema names

If the user identity matches the schema name, the privileges that are associated with schema objects can be given to the user.

Certain privileges associated with schema objects (such as user-defined functions, user-defined distinct types, and stored procedures), can be given if the user identity *matches* the schema name. The schema name is a short SQL identifier used as a qualifier in the name of schema objects and creates a logical grouping of these objects. It is often, but not always, a Db2 authorization ID. For applicable privileges, the RACF access control module looks for a match on schema name before checking RACF profiles.

For authorization checking of the CREATEIN schema privilege, the RACF access control module the RACF access control module first checks to see if the user identity in either of the fields XAPLUCHK or XAPLUPRM matches the schema name in XAPLOBJN. If either of these fields matches XAPLOBJN and XAPLUCHK is not a role, the RACF access control module allows the access. For all other schema privileges, the RACF access control module first checks to see if the user identity in XAPLUCHK matches the schema name in XAPLOWNQ. If those two fields are equal and XAPLUCHK is not a role, the RACF access control module allows the access. In each case, when the RACF access control module allows access, it returns a return code 0 in EXPLRC1 and reason code 14 in EXPLRC2, and no further checking occurs. If the RACF access control module does not allow the access, profile checking occurs. See Chapter 15, “RACF authorization checking reference,” on page 83 for details.

Note: On multilevel-secure systems with the RACF SETROPTS MLS option active, the schema match check is not performed.

If these checks fail, for some privileges the RACF access control module checks whether implicit privileges of ownership from other objects is sufficient.

Implicit privileges of ownership from other objects

The owner of a table space or index space in an implicitly created database has implicit privileges on these objects.

The term *other object* is used to refer to these objects. The owner of the *other object* can be an authorization ID or a role.

Rules for certain database and table space privileges check for ownership of the *other object*. If the *other object* is owned by an authorization ID, the RACF access control module authorizes access and returns a return code 0 in EXPLRC1 and reason code 17 in EXPLRC2. If the *other object* is owned by the role

associated with the user, the RACF access control module authorizes access and returns a return code 0 in EXPLRC1 and reason code 18 in EXPLRC2. For information about which privileges check for ownership of the *other object*, see Chapter 15, “RACF authorization checking reference,” on page 83.

All of the information needed for these checks is included in control block DSNDXAPL which Db2 passes to the RACF access control module. For more information on the fields involved , see “Implicit privileges of ownership” on page 33 and “Implicit privileges of ownership” on page 33.

If these checks fail, profile checking occurs. For details, see Chapter 15, “RACF authorization checking reference,” on page 83.

Logging the Use of Administrative Authorities

The IFCID361 trace record is not written if RACF grants the access due to a administrative authority.

RACF users can specify the AUDIT(SUCCESS) keyword to cause an SMF record to be written when a system authority is used.

Processing cache requests

If Db2 is caching the results of RACF access control module requests, it determines if access is granted for reasons other than the ownership of the object by XAPLUCHK.

Db2 indicates that this type of request is being performed by setting XAPLACAC (XAPLFLG2 bit 5) to '1'B. When this bit is on, and XAPLUCHK is an authorization ID, the RACF access control module suppresses the XAPLUCHK ownership check for the object.

Db2 might set XAPLACAC on the following objects and privileges:

- Package (execute)
- UDF (execute)
- Stored procedure (execute)
- Sequence (usage)
- Table (select, insert, delete, update)
- View (select, insert, delete, update)

View ownership checks on insert, delete, and update are performed against the base table of the view. There is no ownership check for the select privilege on a view.

CREATETMTAB privilege

Access to the CREATETMTAB privilege requires different administrative authorities through RACF access control module.

In Db2, the DBMAINT, DBCTRL, and DBADM administrative authorities are sufficient for the CREATETMTAB privilege. However, with the RACF access control module, a user must have at least one of the following privileges or authorities:

- The CREATETMTAB privilege
- The CREATETAB privilege
- SYSCTRL authority
- SYSADM authority

For the exact class and resource names, see Chapter 15, “RACF authorization checking reference,” on page 83.

CREATE VIEW privilege

If you have sufficient authority, you can create views for other users.

If the installation option DBADM CREATE AUTH on panel DSNTIPP (ZPARM DBACRVW) is set to YES during Db2 installation, users with DBADM authority for "any" database can create views for other users.

When a view is based on tables or a combination of tables and views from more than one database, the view creator must have DBADM for at least one database that contains a table referenced in the view.

The RACF access control module checks the user's DBADM authorization for each database in the list if the XAPLCRVW field indicates that the DBACRVW subsystem parameter is enabled, and the CREATE VIEW privilege is not allowed by the following resources:

- SYSCTRL
- SYSADM
- SYSDBADM

For implicit databases, the check is done on DSNDB04. The result of each DBADM check is placed in the XAPLDBDA field associated with each database.

If a view name is specified with an explicit qualifier, a create view (CRTVUAUTT) authorization check is performed first. Then a check is performed to determine whether the explicit qualifier is a RACF group. If the CRTVUAUTT check fails, RACF issues unauthorized request message ICH408I. If the CRTVUATT check succeeds, and the explicit qualifier is a RACF group, the view is created successfully.

Related concepts

Debugging the RACF access control module

You can use IFCID 0314 trace records to obtain the parameter list on return from the RACF access control module.

CREATE ALIAS privilege

Users with DBADM or DBCTRL privilege for a database can create aliases for other users.

If the installation option DBADM CREATE AUTH on panel DSNTIPP (ZPARM DBACRVW) is set to YES during Db2 installation, users with DBADM or DBCTRL privilege for a database can create aliases for other users.

The RACF access control module checks the user's DBADM and DBCTRL authorization for the database if the XAPLCRVW field indicates that the DBACRVW subsystem parameter is enabled, and the CREATE VEIW privilege is not allowed by the following resources:

- SYSCTRL
- SYSADM
- SYSDBADM

The result of each DBADM and DBCTRL check is placed in the XAPLDBDA field associated with each database.

Related concepts

Debugging the RACF access control module

You can use IFCID 0314 trace records to obtain the parameter list on return from the RACF access control module.

"Any table" privilege

Db2 checks the "any table" privilege for the DESCRIBE TABLE statement.

In Db2, the UPDATE privilege or the REFERENCES privilege for a specific column is sufficient to allow the "any table" privilege. However, when the RACF access control module is invoked, the UPDATE or REFERENCES privilege for a specific column is not sufficient to provide users with the "any table"

privilege. The UPDATE or REFERENCES privilege has to be held on the table to allow the "any table" privilege.

"Any schema" privilege

RACF generic profiles can be used to define protection for sets of similarly named schemas and stored procedures.

RACF does not perform authorization checks looking for "all privileges on all schemas" as Db2 does for the CREATEIN, ALTERIN, DROPIN, and COMMENT ON privileges on schemas; nor does RACF look for "all privileges on all stored procedures" as Db2 does for the EXECUTE privilege for stored procedures. RACF variables and RACF grouping profiles can be used for the protection attributed of schemas and stored procedures that are not similarly named.

UPDATE and REFERENCES authorization on Db2 table columns

You can use the RACF access control module to handle UPDATE and REFERENCES authorizations.

The RACF access control module handles UPDATE and REFERENCES authorizations associated with columns by first checking for access to the entire table (example: *table*.UPDATE) and if not permitted, then to each individual column (example: *table.column*.UPDATE).

When performing an authorization check on a column privilege, the RACF access control module informs Db2 if access is allowed because it is allowed on the whole table or through an individual column. In Db2, this check is performed using fields UPDATECOLS and REFCOLS. The RACF access control module returns a value to Db2 in output field XAPLONWT.

When performing the authorization check on the entire table and authorization is given to the requester, the RACF access control module returns a blank (' ') in the output field XAPLONWT and sends a return code of 0.

If the authorization is given for a particular column or set of columns using a generic profile, the RACF access control module returns an asterisk ('*') in output field XAPLONWT and sends a return code of 0. Db2 provides the column name included in XAPLREL1 to the RACF access control module.

Effect of issuing a PREPARE statement or BIND with the EXPLAIN(ONLY) option when you have the EXPLAIN privilege

The EXPLAIN privilege allows you to prepare and describe SQL statements without having the privileges to execute those SQL statements.

When the RACF access control module is used for authorization checking, and you issue a PREPARE statement, Db2 first checks for the privilege to issue the statement on which the PREPARE is performed. If that privilege check fails, RACF issues message ICH408I. Then Db2 checks for the EXPLAIN privilege. If the EXPLAIN privilege check fails, RACF issues message ICH408I again. If the EXPLAIN privilege check succeeds, RACF does not issue an additional ICH408I message. The statement that Db2 prepares when you have only the EXPLAIN privilege cannot be executed. However, Db2 provides EXPLAIN information for the statement.

Similarly, when the RACF access control module is used for authorization checking, and you bind a package with the EXPLAIN(ONLY) option, Db2 first checks whether you have the privilege to execute the SQL statements in the package. If a privilege check fails, RACF issues unauthorized request message ICH408I. Then Db2 checks whether you have the EXPLAIN privilege. If the EXPLAIN privilege check fails, RACF issues message ICH408I again. If the EXPLAIN privilege check succeeds, BIND processing succeeds, and Db2 provides EXPLAIN information for the statements in the package.

Related reference

Security Server RACF Messages and Codes

Db2 object classes that include privileges in RACF resource class MDSNSM

Some RACF security scenarios include profiles in the MDSNSM resource class for object-level privileges.

If the MDSNSM resource class profile is defined, it might impact the RACF exit return code for object-level profile checks.

For example, suppose that the following conditions exist:

- A profile is defined for SQLADM in the MDSNSM resource class with a universal access authority of NONE:

```
RDEF MDSNSM DB2A.SQLADM UACC(NONE)
```

- No object level profiles are defined for SELECT access on table SYSIBM.SYSTABLES in the MDSNTB class.
- The MDSNTB class has been activated.
- The DSNADM class allows access to certain users, but not USER01.

Now suppose that user USER01 issues the following SELECT statement:

```
SELECT * FROM SYSIBM.SYSTABLES WHERE NAME='CUSTOMER';
```

The RACF exit return code is 8 because the MDSNSM resource class profile is considered to be at the same level as the object profile. If the SQLADM profile were not defined, the RACF exit return code would be 4.

The following table lists the Db2 object-level privileges for which the RACF exit return code can change when privileges are included in RACF resource class MDSNSM.

Table 13. Db2 object-level privileges that include privileges in the MDSNSM resource class

Db2 object type (XAPLTYPE)	Authority or privilege needed in the MDSNSM resource class (XAPLPRIV)
Database (D)	• DISPLAYDB (DSPDBAUTD) • Run REPAIR utility (DIAGAUTD) • STATS (STATSAUTD)
Package (K)	• BIND (BINDAUTK) • COPY (COPYAUTK) • EXECUTE (CHKEXECK) for system-defined routine packages
Plan (P)	BIND (BINDAUTP)
Stored procedure (O)	EXECUTE (CHKEXECO) on system-defined routines
User-defined function (F)	EXECUTE (CHKEXECF) on system-defined routines
Table (T)	• SELECT (SELCTAUTT), INSERT (INSRTAUTT), UPDATE (UPDTEAUTT), DELETE (DELETAUTT), UNLOAD (ULOADAUTT) on catalog and directory tables • Any of the table privileges (ANYTBAUTT), for DESCRIBE TABLE
View (V)	Any of the table privileges (ANYTBAUTV), for DESCRIBE TABLE on a view

Related concepts

FASTAUTH return code translation
Each time the RACF access control module is started, it can also start RACROUTE REQUEST=FASTAUTH multiple times.

The XAPLDIAG output parameter

The output parameter XAPLDIAG is used to contain return codes and reason codes.

When a RACROUTE REQUEST=FASTAUTH check fails to grant access, the RACF access control module records the failing SAF return code, RACF return code, and RACF reason code in XAPLDIAG. Each word of XAPLDIAG contains a FASTAUTH SAF return code (1 byte), the RACF return code (1 byte) and the RACF reason code (2 bytes), from left to right. All return codes and reason codes are shown in hexadecimal. In this way, Db2 or other programs have a way to trap and obtain diagnostic information.

See Chapter 8, “Debugging the RACF access control module,” on page 31 for more information.

Db2 aliases for system-directed access

RACF applies protection to the base object, not to a Db2 alias.

Db2 authorization checks are made using the base object name, not the alias. By the time the RACF access control module is passed the object name, it has already been resolved from the alias name to the base name.

Considerations for remote and local resources

The RACF entity check is always performed for local resources.

Remote resources are always checked by the remote Db2. This also occurs when binding an application that accesses remote resources.

Db2 GRANT statements

The RACF access control module does not call RACF for Db2 GRANT statement checking.

The RACF access control module provides RACF authorization checking of all privileges for all Db2 objects listed in “Privilege names” on page 26 When RACF is called by the RACF access control module, it does not use Db2 authorizations given using Db2 GRANT statements but uses only the resources you defined to RACF.

Structured Query Language (SQL) allows authorities to be held with the WITH GRANT option, which allows users to GRANT those privileges to others. The RACF access control module does not provide this support.

SQL supports the GRANT ALL privilege for any Db2 object. When you use the RACF access control module, you can issue a generic RACF **PERMIT** command to provide the equivalent support. The following command authorizes a user to all Db2 privileges on a Db2 table.

Example:

```
PERMIT Db2-subsystem.table-qualifier.table-name.* CLASS (MDSNTB)
ID(userid) ACCESS(READ)
```

Db2 object names with blank characters

In Db2, it is possible to use delimited identifiers to create Db2 object names containing blank characters.

However, RACF resource names cannot contain blank characters. As a result, when the RACF access control module encounters a Db2 object name containing blank characters, it translates the blank characters to underscores (_, X'6D') before performing security checking. To protect Db2 objects

containing blanks, you must define RACF profiles that match an underscore (either explicitly or with generics) in place of the blank characters.

Related concepts

SQL identifiers (Db2 SQL)

Naming conventions (Db2 SQL)

Db2 object names with special characters

You can use any character that exists in the UTF-8 character set to create a Db2 object name.

Not all of these characters can be represented by the EBCDIC syntactic character set. To protect Db2 objects containing these characters (or any other characters that are not allowed by the RACF command processors, such as commas, semicolons, and parentheses), define RACF profiles containing generic characters to match the unsupported characters.

Exception: The Db2 role object is an exception. Because it is not represented by a RACF profile, the role name can contain characters that are not allowed in a RACF profile name. The choice of a SQL role name must be one that is acceptable to Db2 and RACF. RACF support for SQL roles does not recognize generic characters.

Db2 object names in mixed case

Db2 allows mixed-case object names. However, the mixed-case support in RACF profile names depends on whether IBM-supplied or installation-defined RACF resource classes are used.

- If you use IBM-supplied default RACF resource classes, use generic characters in the RACF profile names to match characters that are in lower case.
- If you use installation-defined RACF resource classes, define the classes with the CASE ASIS option, or use generic characters in the RACF profile names to match characters that are in lower case.

Authority checking for all packages in a collection

You can perform authority checking on a collection of packages instead of performing authority checking on each package individually.

The naming convention for Db2 package objects is:

```
subsystem-name.collection-ID.package-ID.privilege-name
```

When a Db2 user tries to perform an operation on all packages in a collection, Db2 can pass an asterisk (*) to the RACF access control module in place of *package-ID*. To ensure consistent results between the RACF access control module and the RACF command processors (SEARCH and RLIST), the asterisk (*) in the resource name should match the asterisk (*) in the profile name.

For example, in Db2, you can BIND a plan using all of the packages from a given collection. When that plan is later executed, Db2 checks the user's authority to execute all packages in the collection by passing an asterisk (*) in place of the package name. For example, suppose the following Db2 commands are issued for subsystem DSN:

```
BIND PACKAGE(DSNTEP2) MEMBER(DSNTEP2) ACT(REP) ISO(CS)
BIND PLAN(DSNTEP42) PKLIST(DSNTEP2.*) ACT(REP) ISO(CS)
RUN  PROGRAM(DSNTEP2) PLAN(DSNTEP42) -
```

When Db2 gets to the execution step, it calls the RACF access control module to check the user's authority to EXECUTE package DSNTEP2.*, where the asterisk (*) means all packages in the collection.

The RACF access control module checks the user's authority to resource:

```
DSN.DSNTEP2.*.EXECUTE     (in class MDSNPK)
```

The RACF profile name protecting this resource should contain a single asterisk (*) to match the asterisk (*) in the resource name.

Identity used for authorization checks

The RACF access control module receives user identification information in the XAPL (DSNDXAPL) parameter list that is passed by Db2.

In the XAPL, the RACF access control module receives:

- A pointer to the input ACEE that represents the identity of the requester (XAPLUPRM).
- The 1–8-character user ID of the requester (XAPLUPRM).

  Note: The XAPLUPRM value is used for all RACF authorization checking, although RACF actually checks the input ACEE itself to determine this identity. The identity represented by the ACEE is the same as the user ID passed in XAPLUPRM.
- The 1–128-character authorization ID (XAPLUCHK) that Db2 uses for the authorization check. The XAPLUCHK can contain a value that is not a RACF user ID or group, and it can differ from the XAPLUPRM.

While the RACF access control module uses the XAPLUCHK and XAPLUPRM values to perform ownership checks, it performs all access authorization checks using only XAPLUPRM.

It is possible for the XAPLUCHK value to be different from the user ID (XAPLUPRM) represented in the ACEE pointed to by XAPLACEE. For example, this can occur when a BIND request is issued and the binder is not the owner of the plan or package. The RACF access control module is invoked to determine whether the binder is authorized to do the BIND. If this check is successful, it is then invoked to check the binder's authorization to access each Db2 resource accessed in the plan or package. For the BIND check, XAPLUPRM and XAPLUCHK have the authorization ID of the binder. However, for the subsequent checks on the Db2 resources accessed in the plan or package, XAPLUPRM still has the authorization ID of the binder, but XAPLUCHK now has the authorization ID of the plan or package owner. For the BIND to succeed, the binder must have authorization to bind this plan or package, and be authorized to access all Db2 resources accessed in it. Db2 authorization performs the subsequent checks on the owner of the plan/package and not the binder.

AUTHEXIT_CHECK subsystem parameter

If the AUTHEXIT_CHECK subsystem parameter is set to DB2, Db2 provides the ACEE for XAPLUCHK for subsequent checks on the Db2 resources that are accessed in the package. The package owner, not the binder, is checked for authorization.

Related reference

AUTH EXIT CHECK (AUTHEXIT_CHECK subsystem parameter) (Db2 Installation and Migration)

When Db2 cannot provide an ACEE

Db2 cannot provide an ACEE in some situations.

If you are not using external security in CICS (for example, SEC=NO is specified in the DFHSIT), CICS does not pass an ACEE to the CICS attachment facility. When Db2 does not have an ACEE, it passes zeros in the XAPLACEE field. If this happens, your routine can return a 4 in the EXPLRC1 field, and let Db2 handle the authorization check.

Restrictions:

- An ACEE address may not be available for IMS transactions unless IMS is configured to use either APPC/OTMA security full or the IMS Build Security Environment exit (DFSBSEX0). You need to code DFSBSEX0 to return RC4 in register 15, which will instruct IMS to create the ACEE in the dependent region.

- The ACEE address is passed for CICS transactions, when available. If you implement the Db2 CICS attachment facility and CICS is configured to use an external security manager, such as RACF, Db2 passes the ACEE address, if available.
- The ACEE address is passed for Db2 commands, when available. If the master console is used, Db2 does not pass the ACEE address because an ACEE is not available. However, if the user signs on to an MVS operator console, Db2 passes the ACEE address, if available.

Authorization ID, ACEE relationship

XAPL has two authorization ID fields, XAPLUPRM, and XAPLUCHK.

XAPLUPRM is the primary authorization ID and XAPLUCHK is the authorization ID that Db2 uses to perform the authorization. These two fields might have different values.

The ACEE passed in XAPLACEE is that of the primary authorization ID, XAPLUPRM.If XAPLOWAC is on, the ACEE passed in XAPLACEE is that of the authorization ID that Db2 uses to perform the authorization checking, XAPLUCHK.

Invalid or inoperative packages

In Db2, when a privilege required by a package is revoked, the package is invalidated.

If you use an authorization access control routine, it cannot tell Db2 that a privilege is revoked. Therefore, Db2 cannot know to invalidate the package.

If the revoked privilege was EXECUTE on a user-defined function, Db2 marks the package inoperative instead of invalid.

If a privilege that the package depends on is revoked, and if you want to invalidate the package or make it inoperative, you must use the SQL GRANT statement to grant the revoked privilege and then use the REVOKE statement to revoke it. Or, you can set the AUTHEXIT_CACHEREFRESH system parameter to ALL. See AUTH EXIT CACHE REFR (AUTHEXIT_CACHEREFRESH subsystem parameter) (Db2 Installation and Migration) and Invalid and inoperative packages (Managing Security) for more information.

Dropping views

In Db2, when a privilege required to create a view is revoked the view is dropped.

About this task

Like the revocation of plan privileges, such an event is not communicated to Db2 by the authorization checking routine.

If you want Db2 to drop the view when a privilege is revoked, use the SQL statement DROP VIEW.

Caching of EXECUTE on plans

The results of authorization checks on the EXECUTE privilege are not cached when those checks are performed by the exit routine.

Caching of EXECUTE on packages and routines

You can enable package and routine authorization caching on your system.

The results of authorization checks on the EXECUTE privilege for packages and routines are cached. If this privilege is revoked in the exit routine, the cached information is not updated to reflect the revoke. You must use the SQL GRANT and REVOKE statements to update the cached information.

RACF security considerations for caching of dynamic SQL statements

Dynamic statements can be cached when they have passed the authorization checks.

If dynamic statement caching is enabled on your system, dynamic statements can be cached when they have passed the authorization checks. If the privileges that this statement requires are revoked from the authorization ID that is cached with the statement, then this cached statement must be invalidated. If the privilege is revoked in the exit routine this does not happen, and you must use the SQL statements GRANT and REVOKE to refresh the cache.

Resolution of user-defined functions

The create timestamp for the user-defined function must be older than the bind timestamp for the package or plan in which the user-defined function is invoked. If Db2 authorization checking is in effect, and Db2 performs an automatic rebind on a plan or package that invokes a user-defined function, any user-defined functions that were created after the original BIND or REBIND of the invoking plan or package are not candidates for execution.

If you use an access control authorization exit routine, some user-defined functions that were not candidates for execution before the original BIND or REBIND of the invoking plan or package might become candidates for execution during the automatic rebind of the invoking plan or package. If a user-defined function is invoked during an automatic rebind, and that user-defined function is invoked from a trigger body and receives a transition table, the form of the invoked function that Db2 uses for function selection includes only the columns of the transition table that existed at the time of the original BIND or REBIND of the package or plan for the invoking program.

Setting up profiles for Db2 roles

You can use Db2 roles with the RACF access control module.

About this task

Before you can use Db2 roles with the RACF access control module, the security administrator must define RACF profiles to give users access to RACF-protected resources when they are using a role. For example, suppose that you have defined a Db2 trusted context and associated the role TELLER with it. The user ID RANDY is authorized to use the trusted context. You want Randy to have READ access to the resource DSN.PEGGY.TAB.ALTER when he is using the role TELLER.

- Assume that the RACF access control module is configured for multiple subsystem scope. Give RANDY READ authority to the resource DSN.PEGGY.TAB.ALTER when he is using the role TELLER:

```
RDEFINE MDSNTB DSN.PEGGY.TAB.ALTER UACC(NONE)
PERMIT DSN.PEGGY.TAB.ALTER CLASS(MDSNTB) ID(RANDY) ACCESS(READ)
  WHEN(CRITERIA(SQLROLE(TELLER)))
```

  The case of the criteria value (TELLER) is important - it must be entered as it will appear in the CRITERIA parameter of RACROUTE REQUEST=FASTAUTH.

- Make your resource changes take effect:
  - If the class in which you defined the profile is active, refresh the in-storage profiles with your changes:

```
SETROPTS RACLIST(MDSNTB) REFRESH
```

  - If the class in which you defined the profile is not active, stop the Db2 subsystem, activate and RACLIST the class, and restart the Db2 subsystem.

Related concepts

Roles in a trusted context (Managing Security)

Trusted contexts (Managing Security)

Related tasks

If the class was not active

When you define new RACF resources to protect Db2 objects, you must ensure that the new resource definitions become effective.

CREATE and BIND processing

The RACF access control module manages access differently for CREATE and BIND processing.

During CREATE and BIND processing, the RACF access control module grants access only if the user-associated role is on the access list. The role that is associated with the user is contained in XAPLUCHK. These cases occur when XAPLCHKS is OFF.

Initialization

Db2 passes one of three function codes to the RACF access control module for initialization, authorization checking, or termination.

To indicate the function to be performed, Db2 passes one of three function codes to the RACF access control module for initialization, authorization checking, or termination. For general information about initialization and termination information, see Chapter 1, “Introduction to the RACF access control module,” on page 1.

Any Db2 classes you want to use must be active during RACF access control module initialization (XAPLFUNC=1). You cannot activate a Db2 class later and expect the RACF access control module to perform authorization checking against it, because the class will not be RACLISTed. RACLISTing is only done during initialization of the RACF access control module.

To start using Db2 classes that were not previously RACLISTed during initialization, you must stop and restart Db2.

Once the Db2 subsystem has initialized, the following command must be issued to affect profile changes for classes being used by the RACF access control module:

```
SETROPTS RACLIST(classname) REFRESH
```

The following informational messages are issued for each initialization: IRR908I, IRR909I, IRR910I, and IRR911I.

Note: The classes listed in message IRR911I might be a valid subset of the classes listed in message IRR910I. The RACF access control module is programmed to RACLIST all supported Db2 classes. Message IRR910I lists the Db2 classes for which the RACF access control module has initiated RACLIST. However, message IRR911I lists only the Db2 classes that were successfully RACLISTed. In order to be successfully RACLISTed, a Db2 class must be active and contain at least one profile. Therefore, there are valid circumstances where the list of classes contained in IRR911I will be a subset of those listed in IRR910I.

Failure to initialize

If the RACF access control module fails to initialize for any reason, messages IRR900A, IRR901A, IRR902A, and IRR903A are issued to the security console.

If initialization fails, perform the following actions:

1. Check that the Db2 classes are active, and that there is at least one profile defined in each class.
2. Examine RACROUTE REQUEST=LIST return and reason codes to determine why RACLISTing of classes is failing in the RACF access control module.
3. Check if any other required resources (GETMAIN, for example) are obtainable.

Return codes and reason codes from initialization

Return codes from the RACF access control module are returned in the Db2-supplied EXPL field that is called EXPLRC1.

Reason codes from the RACF access control module are returned in the Db2-supplied EXPL field EXPLRC2. See Chapter 12, "XAPLFUNC reference," on page 67 for the meanings of the return and reason codes from the initialization of the RACF access control module.

Deferring to native Db2 authorization

Deferring to native Db2 authorization might require removal of the RACF access control module.

A return code of 4 from the RACF access control module indicates that Db2 defers to Db2 security checking for that particular authorization check.

Removing the RACF access control module

If the RACF access control module is removed, Db2 reverts to using native Db2 authorization. With native Db2 authorization, authority is determined by the Db2 catalogs.

In addition, you might need to inactivate any classes related to the Db2 processing and make the necessary grants in Db2.

Common problems and considerations

If you define special classes in the class descriptor table, you might encounter some common problems.

Common problems that could occur as a result of defining special classes in the class descriptor table (CDT) follow:

- A class is not defined in the CDT.

  This results in a return code of 4 (profile not found) from the RACF access control module.
- If a class is defined in the static CDT, there are incorrect linkage editor procedures from the CDT.
- If a class is defined in the static CDT, it is link-edited properly but a re-IPL has not occurred to pick up the changes.
- If a class is defined in the dynamic CDT, the CDTINFO class was not RACLISTed or refreshed to pick up the changes.
- Single-subsystem scope class names are being used and a new subsystem is using the RACF access control module before classes for the subsystem have been defined.
- Messages IRR900A, IRR901A, IRR902A, and IRR903A are issued because the RACF access control module cannot initialize correctly.

  1. Check to see if Db2 classes are active.
  2. Determine if and why RACLISTing of classes is failing in the module by examining RACROUTE REQUEST=LIST return and reason codes.
  3. Check to see if any other required resources (such as GETMAIN, for example) are obtainable.

Chapter 11. Scenario: Securing data access with RACF facilities at Spiffy Computer

The scenario describes a simple approach for implementing RACF security at Spiffy Computer Company.

This scenario assumes that you have already taken performed the following actions:

- Performed all actions in Scenario: Securing data access with Db2 facilities at Spiffy Computer (Managing Security)
- Installed the RACF access control module

You should base your security plan, techniques, and procedures on your actual security objectives; do not view this sample security plan as an exact model for your security needs. Instead, use it to understand various possibilities and address problem areas that you might encounter when you move from Db2 security to RACF security.

Securing manager access to employee data with RACF

After implementation of RACF security, managers must be able to read data for their employees on the local system or from a remote system.

Specifically, the Spiffy security plan imposes the following security restrictions on managers:

- Managers can retrieve, but not change, all information in the employee table for members of their own departments.
- Managers of managers have the same privileges for their own departments and for the departments that directly report to them.

Creating a RACF group for managers and adding managers to the group

As a first step in giving managers the RACF SELECT privilege on the DEPTMGR table, you need to create a RACF group, and add the managers to it.

Procedure

1. Define a RACF group for the managers.
2. Add manager IDs to the group.

Example

To add user MGROWNER, add RACF group MGRS with MGROWNER as the owner, and add user ID USRT006 to MGRS, use the following statements:

```
ADDUSER  MGROWNER  CLAUTH(DSNR USER) UACC(NONE)
ADDGROUP MGRS SUPGROUP(SYS1) OWNER(MGROWNER)
CONNECT MGROWNER GROUP(MGRS) AUTHORITY(JOIN) UACC(NONE)
ALTUSER MGROWNER DFLTGRP(MGRS)
CONNECT USRT006 GROUP(MGRS)
```

Granting managers the SELECT privilege with RACF security

To provide the SELECT privilege on the DEPTMGR view to managers, you need to permit RACF read access to a profile that defines the SELECT privilege.

Procedure

1. Define a discrete RACF profile for the SELECT privilege on the employee view for managers, DEPTMGR, in the class for views, MDSNTB, with a default access of no access.

2. Permit access to the RACF profile to individual managers.
3. Refresh the profiles in the MDSNTB class.

Example

To define the RACF profile for the SELECT privilege on the DEPTMGR view in subsystem DB2A, grant access to the MGRS group, and refresh the profiles, use the following statements:

```
RDEFINE MDSNTB DB2A.SYSADM.DEPTMGR.SELECT UACC(NONE)
PERMIT DB2A.SYSADM.DEPTMGR.SELECT CLASS(MDSNTB) ID(MGRS) ACC(READ)
SETROPTS RACLIST(MDSNTB) REFRESH
```

Planning for distributed access using RACF security

The Spiffy security planners have determined how the managers can securely access employee data in a distributed environment.

About this task

The Spiffy security plan imposes the following restrictions for distributed access:

- IDs that are managed at the central location hold privileges on views for departments that are at remote locations. For example, the ID MGRD11 has the SELECT privilege on the view DEPTD11.
- If the manager of Department D11 uses a remote system, the ID at that system must be translated to MGRD11. Then a request is sent to the central system. All other IDs are translated to CLERK before they are sent to the central system.
- The communications database (CDB) manages the translated IDs, like MGRD11.
- An ID from a remote system must be authenticated on any request to the central system.

The processes for distributed access at the central server and distributed access at remote locations are the same as the processes for Db2 security.

Related tasks

Implementing distributed access at the central server (Managing Security)
Implementing distributed access at remote locations (Managing Security)

Securing access to payroll operations and management with RACF

After implementation of RACF security, restrictions on how members of the payroll operations department access and handle sensitive payroll information must be unchanged.

The plan imposes the following restrictions on members of the payroll operations department:

- Members of the payroll operations department can update any column of the employee table except for SALARY, BONUS, and COMM.
- Members of payroll operations can update any row except for rows that are for members of their own department.

Because changes to the table are made only from the central location, distributed access does not affect payroll operations.

Views of payroll operations, and methods of securing compensation data are the same as those that are used when DB2 security is used.

Related tasks

Creating views of payroll operations (Managing Security)
Securing compensation accounts with update tables (Managing Security)
Securing compensation updates with other measures (Managing Security)

Creating a RACF group for access to payroll data and adding payroll operations workers to the group

As a first step in giving payroll operations workers the RACF SELECT, INSERT, UPDATE, and DELETE privileges on the PAYDEPT view, you need to create a RACF group, and add the IDs of users who can have access to payroll data to the group.

Procedure

1. Define a RACF group for the payroll personnel.
2. Add payroll personnel IDs to the group.

Example

To add user PAYOWNER, add RACF group PAYOPS with PAYOWNER as the owner, and add user ID USRT010 to PAYOPS, use the following statements:

```
ADDUSER  PAYOWNER  CLAUTH(DSNR USER) UACC(NONE)
ADDGROUP PAYOPS SUPGROUP(SYS1) OWNER(PAYOWNER)
CONNECT PAYOWNER GROUP(PAYOPS) AUTHORITY(JOIN) UACC(NONE)
ALTUSER PAYOWNER DFLTGRP(PAYOPS)
CONNECT USRT010 GROUP(PAYOPS)
```

Granting RACF access to payroll operations to a RACF group

To provide the SELECT, INSERT, UPDATE, and DELETE privileges on the PAYDEPT view to payroll workers, you need to permit RACF read access to a profile that defines the SELECT, INSERT, UPDATE, and DELETE privileges.

Procedure

1. Define a discrete RACF profile for the SELECT, INSERT, UPDATE, and DELETE privileges on the view for payroll workers, PAYDEPT, in the class for views, MDSNTB, with a default access of no access.
2. Permit access to the RACF profile to individual managers.
3. Refresh the profiles in the MDSNTB class.

Example

To define the RACF profiles for the SELECT, INSERT, UPDATE, and DELETE privileges on the PAYDEPT table in subsystem DB2A, and grant access to the PAYOPS group, use the following statements:

```
RDEFINE MDSNTB DB2A.SYSADM.PAYDEPT.SELECT UACC(NONE)
RDEFINE MDSNTB DB2A.SYSADM.PAYDEPT.INSERT UACC(NONE)
RDEFINE MDSNTB DB2A.SYSADM.PAYDEPT.UPDATE UACC(NONE)
RDEFINE MDSNTB DB2A.SYSADM.PAYDEPT.DELETE UACC(NONE)
PERMIT DB2A.SYSADM.PAYDEPT.SELECT CLASS(MDSNTB) ID(PAYOPS) ACC(READ)
PERMIT DB2A.SYSADM.PAYDEPT.INSERT CLASS(MDSNTB) ID(PAYOPS) ACC(READ)
PERMIT DB2A.SYSADM.PAYDEPT.UPDATE CLASS(MDSNTB) ID(PAYOPS) ACC(READ)
PERMIT DB2A.SYSADM.PAYDEPT.DELETE CLASS(MDSNTB) ID(PAYOPS) ACC(READ)
SETROPTS RACLIST(MDSNTB) REFRESH
```

Creating a RACF group for payroll managers and adding payroll managers to the group

When you implemented Db2 managed security for the Spiffy database, you created a RACF group for payroll managers. You need to add the IDs of managers who can have access to payroll data to the group.

Procedure

1. If you have not already done so, define a RACF group for the payroll managers.

2. Add payroll manager IDs to the group.

Example

To add user PAYMGR, add RACF group PAYMGRS with PAYMGR as the owner, and add user ID USRT020 to PAYMGRS, use the following statements:

```
ADDUSER  PAYMGR  CLAUTH(DSNR USER) UACC(NONE)
ADDGROUP PAYMGRS SUPGROUP(SYS1) OWNER(PAYMGR)
CONNECT PAYMGR GROUP(PAYMGRS) AUTHORITY(JOIN) UACC(NONE)
ALTUSER PAYMGR DFLTGRP(PAYMGRS)
CONNECT USRT020 GROUP(PAYMGRS)
```

Granting RACF access for payroll management to a RACF group

During implementation of RACF security for the Spiffy database, RACF profiles for access to the PAYMGR view must be created, and access to those profiles must be granted to the PAYMGRS RACF group.

About this task

The security administrator associates the payroll managers' IDs with the PAYMGRS group. Next, privileges on the PAYMGR view, the compensation application, and the payroll update application are granted to PAYMGRS. The payroll update application must have the appropriate privileges on the update table.

Example

Suppose that ID SYSADM created the PAYMGR view in subsystem DB2A. To define the RACF profiles for the SELECT, INSERT, UPDATE, and DELETE privileges on the PAYMGR view in subsystem DB2A, and grant access to the PAYMGRS group, use statements like these:

```
RDEFINE MDSNTB DB2A.SYSADM.PAYMGR.SELECT UACC(NONE)
RDEFINE MDSNTB DB2A.SYSADM.PAYMGR.INSERT UACC(NONE)
RDEFINE MDSNTB DB2A.SYSADM.PAYMGR.UPDATE UACC(NONE)
RDEFINE MDSNTB DB2A.SYSADM.PAYMGR.DELETE UACC(NONE)
PERMIT DB2A.SYSADM.PAYMGR.SELECT CLASS(MDSNTB) ID(PAYMGRS) ACC(READ)
PERMIT DB2A.SYSADM.PAYMGR.INSERT CLASS(MDSNTB) ID(PAYMGRS) ACC(READ)
PERMIT DB2A.SYSADM.PAYMGR.UPDATE CLASS(MDSNTB) ID(PAYMGRS) ACC(READ)
PERMIT DB2A.SYSADM.PAYMGR.DELETE CLASS(MDSNTB) ID(PAYMGRS) ACC(READ)
SETROPTS RACLIST(MDSNTB) REFRESH
```

Suppose that the application plan name for the compensation application is COMPENS. To define a RACF profile for the EXECUTE privilege on the compensation application, and grant access to the PAYMGRS group, use statements like these:

```
RDEFINE MDSNPN DB2A.COMPENS.EXECUTE UACC(NONE)
PERMIT DB2A.COMPENS.EXECUTE CLASS(MDSNPN) ID(PAYMGRS) ACC(READ)
SETROPTS RACLIST(MDSNPN) REFRESH
```

Managing access privileges of other authorities with RACF security

In addition to the privileges for the managers and the payroll operation and management personnel, the security plan considers the privileges for other roles.

Creating a RACF group for database administrators and adding database administrators to the group

You need to create a RACF group for users who need database administration authority on the Spiffy database.

Procedure

1. Define a RACF group for database administrators.

2. Add IDs of database administrators to the group.

Example

To add user DBAOWNER, add RACF group DB2ADMIN with DBAOWNER as the owner, and add user ID ADMF010 to DB2ADMIN, use the following statements:

```
ADDUSER  DBAOWNER  CLAUTH(DSNR USER) UACC(NONE)
ADDGROUP DB2ADMIN SUPGROUP(SYS1) OWNER(DBAOWNER)
CONNECT DBAOWNER GROUP(DB2ADMIN) AUTHORITY(JOIN) UACC(NONE)
ALTUSER DBAOWNER DFLTGRP(DB2ADMIN)
CONNECT ADMF010 GROUP(DB2ADMIN)
```

Related tasks

Granting database administration authority to the Spiffy database with RACF
As with the Db2 security plan for the Spiffy database, the RACF security plan requires that the database administrator does not have all the implicit privileges of DBADM authority.

Creating a RACF group for system administrators and adding system administrators to the group
To limit the number of users with system administration privileges, you need to create a RACF group for system administrators at the Spiffy company.

Granting system administration authority with RACF
As with the Db2 security plan, the RACF security planners want to minimize risk by granting the SYSADM authority to as few users as possible.

Managing access by object owners
The Spiffy security plan must consider the ID that owns and grants privileges on the tables, views, and programs. The ID that owns these objects has many implicit privileges on the objects. The owner of the objects can also grant privileges on the objects to other users.

Auditing access with RACF security
To ensure that only intended users have access to Spiffy Computer resources when you use RACF security, you can generate a RACF audit report.

Granting database administration authority to the Spiffy database with RACF

As with the Db2 security plan for the Spiffy database, the RACF security plan requires that the database administrator does not have all the implicit privileges of DBADM authority.

About this task

Using Db2 security facilities, the Spiffy security planners granted the following privileges administrative privileges to RACF group ID DB2ADMIN:

- DBCTRL authority over the DSN8D11A database
- The INDEX privilege on all tables in the database except the employee table and the payroll update table
- The SELECT, INSERT, UPDATE, and DELETE privileges on certain tables, excluding the employee table and the payroll update table

The security administrator needs to grant the same privileges to the DB2ADMIN group using RACF security.

Example

To allow DBCTRL authority on the DSN8D11A database, and the INDEX, SELECT, INSERT, UPDATE, and DELETE privileges on the project table using RACF security, use the following statements:

```
RDEFINE DSNADM DB2A.DSN8D11A.DBCTRL UACC(NONE)
RDEFINE MDSNTB DB2A.DSN8B10.PROJ.INDEX UACC(NONE)
RDEFINE MDSNTB DB2A.DSN8B10.PROJ.SELECT UACC(NONE)
```

```
RDEFINE MDSNTB DB2A.DSN8B10.PROJ.INSERT UACC(NONE)
RDEFINE MDSNTB DB2A.DSN8B10.PROJ.UPDATE UACC(NONE)
RDEFINE MDSNTB DB2A.DSN8B10.PROJ.DELETE UACC(NONE)
PERMIT DB2A.DSN8D11A.DBCTRL CLASS(DSNADM) ID(DB2ADMIN) ACC(READ)
PERMIT DB2A.DSN8B10.PROJ.INDEX CLASS(MDSNTB) ID(DB2ADMIN) ACC(READ)
PERMIT DB2A.DSN8B10.PROJ.SELECT CLASS(MDSNTB) ID(DB2ADMIN) ACC(READ)
PERMIT DB2A.DSN8B10.PROJ.INSERT CLASS(MDSNTB) ID(DB2ADMIN) ACC(READ)
PERMIT DB2A.DSN8B10.PROJ.UPDATE CLASS(MDSNTB) ID(DB2ADMIN) ACC(READ)
PERMIT DB2A.DSN8B10.PROJ.DELETE CLASS(MDSNTB) ID(DB2ADMIN) ACC(READ)
SETROPTS RACLIST(DSNADM) REFRESH
SETROPTS RACLIST(MDSNTB) REFRESH
```

Related tasks

Creating a RACF group for database administrators and adding database administrators to the group
You need to create a RACF group for users who need database administration authority on the Spiffy database.

Creating a RACF group for system administrators and adding system administrators to the group
To limit the number of users with system administration privileges, you need to create a RACF group for system administrators at the Spiffy company.

Granting system administration authority with RACF
As with the Db2 security plan, the RACF security planners want to minimize risk by granting the SYSADM authority to as few users as possible.

Managing access by object owners
The Spiffy security plan must consider the ID that owns and grants privileges on the tables, views, and programs. The ID that owns these objects has many implicit privileges on the objects. The owner of the objects can also grant privileges on the objects to other users.

Auditing access with RACF security
To ensure that only intended users have access to Spiffy Computer resources when you use RACF security, you can generate a RACF audit report.

Managing access by the DBADM authority (Managing Security)

Creating a RACF group for system administrators and adding system administrators to the group

To limit the number of users with system administration privileges, you need to create a RACF group for system administrators at the Spiffy company.

Procedure

1. Define a RACF group for system administrators.
2. Add IDs of system administrators to the group.

Example

To add user DB2OWNER, add RACF group DB2SYSTM with DB2OWNER as the owner, and add user ID ADMF005 to DB2SYSTM, use the following statements:

```
ADDUSER  DB2OWNER  CLAUTH(DSNR USER) UACC(NONE)
ADDGROUP DB2SYSTM SUPGROUP(SYS1) OWNER(DB2OWNER)
CONNECT DB2OWNER GROUP(DB2SYSTM) AUTHORITY(JOIN) UACC(NONE)
ALTUSER DB2OWNER DFLTGRP(DB2SYSTM)
CONNECT ADMF005 GROUP(DB2SYSTM)
```

Related tasks

Creating a RACF group for database administrators and adding database administrators to the group
You need to create a RACF group for users who need database administration authority on the Spiffy database.

Granting database administration authority to the Spiffy database with RACF

As with the Db2 security plan for the Spiffy database, the RACF security plan requires that the database administrator does not have all the implicit privileges of DBADM authority.

Granting system administration authority with RACF
As with the Db2 security plan, the RACF security planners want to minimize risk by granting the SYSADM authority to as few users as possible.

Managing access by object owners
The Spiffy security plan must consider the ID that owns and grants privileges on the tables, views, and programs. The ID that owns these objects has many implicit privileges on the objects. The owner of the objects can also grant privileges on the objects to other users.

Auditing access with RACF security
To ensure that only intended users have access to Spiffy Computer resources when you use RACF security, you can generate a RACF audit report.

Granting system administration authority with RACF

As with the Db2 security plan, the RACF security planners want to minimize risk by granting the SYSADM authority to as few users as possible.

About this task

The security administrator needs to use RACF security to assign system administration privileges to the RACF group that is intended for any users who need those privileges.

Example

To assign SYSADM authority to group DB2SYSTM in the DB2A subsystem using RACF security, use the following statements:

```
RDEFINE DSNADM DB2A.SYSADM UACC(NONE)
PERMIT DB2A.SYSADM CLASS(DSNADM) ID(DB2SYSTM) ACC(READ)
SETROPTS RACLIST(DSNADM) REFRESH
```

Related tasks

Creating a RACF group for database administrators and adding database administrators to the group
You need to create a RACF group for users who need database administration authority on the Spiffy database.

Granting database administration authority to the Spiffy database with RACF
As with the Db2 security plan for the Spiffy database, the RACF security plan requires that the database administrator does not have all the implicit privileges of DBADM authority.

Creating a RACF group for system administrators and adding system administrators to the group
To limit the number of users with system administration privileges, you need to create a RACF group for system administrators at the Spiffy company.

Managing access by object owners
The Spiffy security plan must consider the ID that owns and grants privileges on the tables, views, and programs. The ID that owns these objects has many implicit privileges on the objects. The owner of the objects can also grant privileges on the objects to other users.

Auditing access with RACF security
To ensure that only intended users have access to Spiffy Computer resources when you use RACF security, you can generate a RACF audit report.

Managing access by the SYSADM authority (Managing Security)

Managing access by object owners

The Spiffy security plan must consider the ID that owns and grants privileges on the tables, views, and programs. The ID that owns these objects has many implicit privileges on the objects. The owner of the objects can also grant privileges on the objects to other users.

About this task

The Spiffy security planners want to limit the number of IDs that have privileges on the employee table and the payroll update table to the smallest convenient value. To meet that objective, they decide that the owner of the employee table should issue all of the CREATE VIEW and GRANT statements. They also decide to have the owner of the employee table own the plans and packages that are associated with employee data. The employee table owner implicitly has the following privileges, which the plans and packages require:

- The owner of the payroll update program must have the SELECT privilege on the payroll update table and the UPDATE privilege on the employee table.
- The owner of the commission program must have the UPDATE privilege on the payroll update table and the SELECT privilege on the employee table.

The owners of several other payroll programs must have the proper privileges to do payroll processing, such as printing payroll checks, writing summary reports, and so on.

To bind these plans and packages, an ID must have the BIND or BINDADD privileges. The list of privileges that are required by the owner of the employee table suggests the functional approach. The Spiffy security planners create a RACF group for the owner of the employee table.

Related tasks

Creating a RACF group for database administrators and adding database administrators to the group
You need to create a RACF group for users who need database administration authority on the Spiffy database.

Granting database administration authority to the Spiffy database with RACF
As with the Db2 security plan for the Spiffy database, the RACF security plan requires that the database administrator does not have all the implicit privileges of DBADM authority.

Creating a RACF group for system administrators and adding system administrators to the group
To limit the number of users with system administration privileges, you need to create a RACF group for system administrators at the Spiffy company.

Granting system administration authority with RACF
As with the Db2 security plan, the RACF security planners want to minimize risk by granting the SYSADM authority to as few users as possible.

Auditing access with RACF security
To ensure that only intended users have access to Spiffy Computer resources when you use RACF security, you can generate a RACF audit report.

Auditing access with RACF security

To ensure that only intended users have access to Spiffy Computer resources when you use RACF security, you can generate a RACF audit report.

Procedure

1. In RDEFINE commands that define RACF profiles for Db2 resources, include the AUDIT(ALL(READ)) option to direct RACF to write audit information to SMF data sets.

   Example: In the following RDEFINE command, the AUDIT option causes records to be written to an SMF data set when a SELECT operation is performed on the PAYDEPT view.

   ```
   RDEFINE MDSNTB DB2A.SYSADM.PAYDEPT.SELECT UACC(NONE) AUDIT(ALL(READ))
   ```

2. When the SMF recording data sets become full, dump the contents of the recording data sets to permanent data sets using one of the SMF dump utilities, IFASMFDP or IFASMFDL. Dump all types and subtypes of SMF records.

   Example: The following JCL job step formats the contents of SYS1.MANX to sequential data set SYSADM.SMFDATA1.

   ```
   //SMFDUMP EXEC PGM=IFASMFDP
   //SYSPRINT DD SYSOUT=*
   //ADUPRINT DD SYSOUT=*
   //DUMPIN   DD  DSN=SYS1.MANX,DISP=SHR
   //DUMPOUT1 DD  DSN=SYSADM.SMFDATA2,DISP=(NEW,KEEP),UNIT=SYSDA,
   //             SPACE=(CYL,(10,2))
   //SYSIN DD *
   INDD(DUMPIN,OPTIONS(DUMP))
   OUTDD(DUMPOUT2,TYPE(000:255))
   /*
   ```

3. Run IFASMFDP or IFASMFDL with the IRRADU00 exit on the data sets you populated in step “2” on page 65 to retrieve and format the SMF records that are related to RACF access.

   Example: Suppose that SMF records have been stored in SMF data set SYSADM.SMFDATA1. The following JCL job step formats RACF-related records from SYSADM.SMFDATA1 and stores them in sequential data set SMF.UNLOAD1.

   ```
   //SMFDUMP EXEC PGM=IFASMFDP
   //SYSPRINT DD SYSOUT=*
   //ADUPRINT DD SYSOUT=*
   //DUMPIN   DD  DSN=SYSADM.SMFDATA1,DISP=SHR
   //DUMPOUT DD DUMMY
   //OUTDD    DD  DSN=SMF.UNLOAD1,DISP=(NEW,CATLG,DELETE),
   //         SPACE=(CYL,(100,10),RLSE),UNIT=SYSDA,DCB=(LRECL=12288,RECFM=VB)
   //SYSIN DD *
   ABEND(NORETRY)
   USER2(IRRADU00)
   USER3(IRRADU86)
   /*
   ```

4. Optional: Load the contents of the data set that you populated in step “3” on page 65 into Db2 tables so that you can easily retrieve and examine the records of interest.

   See Using the RACF SMF data unload utility output with Db2 for information about sample jobs for creating tables and a sample LOAD statement for loading data into the tables.

Related tasks

Creating a RACF group for database administrators and adding database administrators to the group
You need to create a RACF group for users who need database administration authority on the Spiffy database.

Granting database administration authority to the Spiffy database with RACF
As with the Db2 security plan for the Spiffy database, the RACF security plan requires that the database administrator does not have all the implicit privileges of DBADM authority.

Creating a RACF group for system administrators and adding system administrators to the group
To limit the number of users with system administration privileges, you need to create a RACF group for system administrators at the Spiffy company.

Granting system administration authority with RACF
As with the Db2 security plan, the RACF security planners want to minimize risk by granting the SYSADM authority to as few users as possible.

Managing access by object owners
The Spiffy security plan must consider the ID that owns and grants privileges on the tables, views, and programs. The ID that owns these objects has many implicit privileges on the objects. The owner of the objects can also grant privileges on the objects to other users.

Chapter 12. XAPLFUNC reference

Db2 uses function codes to call the RACF access control module.

The following table shows the purpose and timing of each function call.

Table 14. XAPLFUNC codes and corresponding functions

Function code	Time of call	Purpose
XAPLFUNC=1	Db2 initialization	Create in-storage profiles and indicate what action Db2 must take if the RACF access control module abends or fails to initialize.
XAPLFUNC=2	Db2 authorization	Check Db2 objects and authorities.
XAPLFUNC=3	Db2 termination	Delete in-storage profiles.

Unsupported function codes: If the RACF access control module receives a XAPLFUNC function code other than 1, 2 or 3, the RACF access control module sends a return code of 12 to the caller.

When a return code of 12 is received:

- Native Db2 authorization is used if &ERR0R0PT 1 or the level of Db2 is below DB2 Version 7.
- The Db2 subsystem stops if &ERR0R0PT 2 and the level of Db2 is DB2 Version 7 or later.

Initialization (XAPLFUNC = 1)

When the RACF access control module is called with XAPLFUNC function code of 1, it issues a RACROUTE REQUEST=STAT request to determine if RACF is active.

If RACF is not active, the RACF access control module returns to Db2 with a return code of 12. If RACF is active, the RACF access control module builds the class names, as specified by the assembler SET symbols, and performs a RACROUTE REQUEST=LIST,CLASS=*classname* for each new Db2-related class.

Attention

- If you override &CLASSNMT or use the single-subsystem scope, the RACF access control module uses only installation-defined classes.
- If you use the multiple-subsystem scope with the default &CLASSNMT, the RACF access control module uses classes supplied by IBM.

The RACROUTE REQUEST=LIST,ENVIR=CREATE,GLOBAL=YES request brings profiles to a data space for that particular Db2 or allows a subsequent Db2 to use those in-storage profiles.

If no Db2-related classes were active, a failure occurs and the RACF access control module ends with a return code of 12.

Note: The following are not failures:

- A class is not active (SAF RC=4, RACF RC=10)
- A class is not defined (SAF RC=4, RACF RC=8)

If a class is not active or does not exist for an object or authority, the RACF access control module defers to Db2 for authorization checking and ends with a return code of 4.

If *one* request fails, the *entire* initialization fails. When this happens, the RACF access control module cleans up all the resources and ends with a return code of 12.

If you want to use Db2 classes for authorization against Db2 objects, the classes must be active when the subsystem is started.

Failures during initialization processing are indicated by a return and reason code pair and a message.

Initialization return and reason codes

The following return and reason codes are shown in decimal notation.

Return code
Meaning

0
Initialization successful.

Reason code
Meaning

0
Installation option &ERR0R0PT was set to 1. Therefore, native Db2 authorization is used in the event of an error.

16
Installation option &ERR0R0PT was set to 2. Therefore, the Db2 system is requested to stop in the event of an error on a subsequent authorization check.

12
Initialization unsuccessful; don't call RACF access control module again.

Reason code
Meaning

1
An input Db2 subsystem ACEE was not provided. Installation option &ERR0R0PT was set to 1. Therefore, native Db2 authorization is used.

2
RACF is not active. Installation option &ERR0R0PT was set to 1. Therefore, native Db2 authorization is used.

3
RACROUTE REQUEST=LIST,ENVIR=CREATE failure. Installation option &ERR0R0PT was set to 1. Therefore, native Db2 authorization is used.

4
No active Db2 classes. Installation option &ERR0R0PT was set to 1. Therefore, native Db2 authorization is used.

10
Incorrect XAPL level. The value of XAPLLVL is less than V8R1M0. Installation option &ERR0R0PT was set to 1. Therefore, native Db2 authorization is used.

12
Input Db2 subsystem ACEE was not valid. Installation option &ERR0R0PT was set to 1. Therefore, native Db2 authorization is used. Db2 authorization is used.

16
An initialization error occurred. Installation option &ERR0R0PT was set to 2. Therefore, the Db2 subsystem is requested to stop.

Authorization checking (XAPLFUNC = 2)

The RACF access control module requires an input ACEE to perform authority checking.

When an input ACEE (XAPLACEE) is not provided to the RACF access control module, it defers to Db2 for authority checking (EXPLRC1 set to 4). For the requests for which the input ACEE (XAPLACEE) is set to zero, see “When Db2 cannot provide an ACEE” on page 52. For these requests, authority checking must be implemented using the Db2 GRANT and REVOKE statements. RACF profiles defined for these requests are not used.

The RACF access control module performs FASTAUTH checks during authorization according to the rules described in Chapter 15, “RACF authorization checking reference,” on page 83. In Db2, there is no concept of negative access level. RACF access control module processing ends when FASTAUTH returns a return code of 0 or the list of checks for the request has been exhausted. Failure audit records are only created for the first failing resource. All audit records associated with the same invocation of the RACF access control module contain the same LOGSTR data.

Authorization return and reason codes

The following return and reason codes are shown in decimal notation.

Return code
Meaning

0
Access permitted

Reason code
Meaning

0
Access permitted by FASTAUTH checking.

13
Access permitted by implicit privilege of ownership.

14
Access permitted because current SQL ID matches schema name.

16
Access permitted because the role associated with the request owns the object.

17
Access permitted because the authorization ID associated with the request owns the implicit object.

18
Access permitted because the role associated with the request owns the implicit object.

4
Unable to determine; perform Db2 authorization checking

Reason code
Meaning

0
Input class (XAPLTYPE) not active.

11
Input ACEE (XAPLACEE) not provided.

14
The ALET could not be created for cross memory ACEE.

15
Input privilege code (XAPLPRIV) or input class (XAPLTYPE) not defined to the RACF access control module.

16
Input privilege code (XAPLPRIV) does not contain any rules.

18
Issued when running on z/OS 1.7 and trying to create an object in a trusted context with the "role as object owner" clause.

8
Access denied

Reason code
Meaning

0
Access denied.

17
Autobind indicator (XAPLAUTO) is not zero, indicating an autobind was requested. Manual REBIND is required.

18
DSNXRXAC was assembled with z/OS 1.7 or earlier macros and an authorization check is being made where only a role can allow access.

100
Role information was passed, but ignored because the RACF access control module was assembled with z/OS 1.7 macros.

Related concepts
Authorization processing examples

FASTAUTH return code translation

Each time the RACF access control module is started, it can also start RACROUTE REQUEST=FASTAUTH multiple times.

If one of the FASTAUTH requests is completed with a return code of zero, the return code passed back to Db2 is zero. If none of the FASTAUTH requests are completed with a return code of zero, the collection of return codes from FASTAUTH must be translated into a single resultant return code. Return code translation can be summarized as follows:

If all object resource checks result in a return code of 4 and none of the DSNADM checks result in a return code of 0, the RACF access control module passes back a return code of 4.

If at least one object resource check results in a return code of 8 and none of the DSNADM checks result in a return code of 0, the RACF access control module passes back a return code of 8.

If no object resource profiles are checked and all of the DSNADM checks result in a return code of 8, the RACF access control module passes back a return code of 8. Otherwise, if no object resources are checked and the DSNADM checks result in a mix of 4s and 8s, the RACF access control module passes back a return code of 4.

All failing SAF or RACF return codes and RACF reason codes are placed in the output parameter field in XAPLDIAG, to be returned to Db2. This information is then available to Db2, SQL, or other programs to obtain diagnostic information from it.

The following table illustrates the method used to do this translation.

Table 15. FASTAUTH return code translation

Return code from object profile or system profile in MDSNSM resource class^{"2" on page 71}	Return code from ADM profile	Output return code
—	All 4s	04
—	All 8s	08
—	Mix^{"1" on page 71}	04
All 4s	All 4s	04
All 4s	All 8s	04
All 4s	Mix^{"1" on page 71}	04
All 8s	All 4s	08
All 8s	All 8s	08

Table 15. FASTAUTH return code translation (continued)

Return code from object profile or system profile in MDSNSM resource class"2" on page 71	Return code from ADM profile	Output return code
All 8s	Mix"1" on page 71	08
Mix"1" on page 71	All 4s	08
Mix"1" on page 71	All 8s	08
Mix"1" on page 71	Mix"1" on page 71	08

Note:

1. Mix indicates various 4 and 8 return codes.
2. For details on the Db2 privileges that are in MDSNSM resource classes, see "Db2 object classes that include privileges in RACF resource class MDSNSM" on page 49.

Termination (XAPLFUNC = 3)

When the RACF access control module module uses XAPLFUNC function code 3, it issues a RACROUTE REQUEST=LIST,ENVIR=DELETE,GLOBAL=YES request. The classes that were previously brought into storage during Db2 initialization are deleted.

Failures during termination processing are indicated by a return and reason code pair and a message.

Termination return and reason codes

The following return and reason codes are shown in decimal notation.

Return code
Meaning

0
Termination successful

8
Termination failure

Reason code
Meaning

1
Input Db2 subsystem ACEE was not provided.

7
RACROUTE REQUEST=LIST,ENVIR=DELETE failure.

12
Input Db2 subsystem ACEE was not valid.

Chapter 13. Supplied RACF resource classes for Db2

The following RACF classes for Db2 objects and administrative authorities are supplied in the class descriptor table (CDT).

Table 16. Resource classes for Db2 objects and administrative authorities

Class name	Description
DSNADM	Db2 administrative authority class
DSNR	Controls access to Db2 subsystems
GDSNBP	Grouping class for Db2 buffer pool privileges
GDSNCL	Grouping class for Db2 collection privileges
GDSNDB	Grouping class for Db2 database privileges
GDSNJR	Grouping class for Java archive files (JARs)
GDSNPK	Grouping class for Db2 package privileges
GDSNPN	Grouping class for Db2 plan privileges
GDSNSC	Grouping class for Db2 schemas privileges
GDSNSG	Grouping class for Db2 storage group privileges
GDSNSM	Grouping class for Db2 system privileges
GDSNSP	Grouping class for Db2 stored procedure privileges
GDSNSQ	Grouping class for Db2 sequences
GDSNTB	Grouping class for Db2 table, index, or view privileges
GDSNTS	Grouping class for Db2 tablespace privileges
GDSNUF	Grouping class for Db2 user-defined function privileges
GDSNUT	Grouping class for Db2 user-defined distinct type privileges
MDSNBP	Member class for Db2 buffer pool privileges
MDSNCL	Member class for Db2 collection privileges
MDSNDB	Member class for Db2 database privileges
MDSNJR	Member class for Java archive files (JARs)
MDSNPK	Member class for Db2 package privileges
MDSNPN	Member class for Db2 plan privileges
MDSNSC	Member class for Db2 schema privileges
MDSNSG	Member class for Db2 storage group privileges
MDSNSM	Member class for Db2 system privileges
MDSNSP	Member class for Db2 stored procedure privileges
MDSNSQ	Member class for Db2 sequences
MDSNTB	Member class for Db2 table, index, or view privileges
MDSNTS	Member class for Db2 table space privileges

Table 16. Resource classes for Db2 objects and administrative authorities (continued)

Class name	Description
MDSNUF	Member class for Db2 user-defined function privileges
MDSNUT	Member class for Db2 user-defined distinct type privileges

Chapter 14. Authorization processing examples

- Examples 1 through 4 show authority checks performed on tables using supplied classes for multiple-subsystem scope (&CLASSOPT 2).
- Example 5 shows authority checks performed on tables using installation-defined classes for multiple-subsystem scope (&CLASSOPT 2).
- Example 6 shows authority checks performed on tables using installation-defined classes for single-subsystem scope (&CLASSOPT 1).

Example 1: Allowing access (auditing for failures)

RACF access control module can grant access to Db2 objects based on a Db2 administrative authority profile.

This example shows how the RACF access control module allows access to a Db2 object (a table) based on a Db2 administrative authority profile. Auditing is activated for failures.

In this example, user ID MIKEJ is trying to alter a table called BDA0828.EMP in database JBW2000.

Setup

- Classification model (&CLASSOPT): 2
- Class name root (&CLASSNMT): DSN
- Class name suffix (&CHAROPT): 1

  This is the default value, but it is not used with supplied classes.
- Db2 subsystem name: VHH1
- **Profiles:**
  - Defined in the MDSNTB class:

    VHH1.BDA0828.EMP.ALTER

    - AUDIT(FAILURES(READ))
    - UACC(NONE)
  - Defined in the DSNADM class:

    VHH1.SYSADM

    - AUDIT(FAILURES(READ))
    - UACC(NONE)
    - ID(MIKEJ) ACCESS(READ)
- User ID MIKEJ has SYSADM authority.

Profile checking

RACF checks the following resources:

- VHH1.BDA0828.EMP.ALTER in class MDSNTB

  Results:
  - Access is denied (return code 8).
  - No failure message (ICH408I) is issued.
  - No audit records are created.
- VHH1.JBW2000.DBADM in class DSNADM

Results:

- No profile is found (return code 4).
- No failure message (ICH408I) is issued.
- No audit records are created.

- VHH1.SYSADM in class DSNADM

  Results:

  - Access is granted (return code 0).
  - No failure message (ICH408I) is issued.
  - No audit records are created.

Final result

The RACF access control module sends a return code of 0 to Db2.

Example 2: Allowing access (auditing for all attempts)

You can use the RACF access control module to grant access to Db2 objects.

This example shows how the RACF access control module allows access to a Db2 object (a table) based on a Db2 administrative authority profile. Auditing is activated for all access attempts.

In this example, user ID MIKEJ is trying to alter a table called BDA0828.EMP in database JBW2000.

Setup

- Classification model (&CLASSOPT): 2
- Class name root (&CLASSNMT): DSN
- Class name suffix (&CHAROPT): 1

  This is the default value, but it is not used with supplied classes.
- Db2 subsystem name: VHH1
- **Profiles:**
  - Defined in the MDSNTB class:

    VHH1.BDA0828.EMP.ALTER

    ```
    - AUDIT(ALL(READ))
    - UACC(NONE)
    - ID(MIKEJ) ACCESS(NONE)
    ```
  - Defined in the DSNADM class:

    VHH1.SYSADM

    ```
    - AUDIT(ALL(READ))
    - UACC(NONE)
    - ID(MIKEJ) ACCESS(READ)
    ```
- User ID MIKEJ has SYSADM authority.

Profile checking

RACF checks the following resources:

- VHH1.BDA0828.EMP.ALTER in class MDSNTB

  Results:

  - Access is denied (return code 8).
  - No failure message (ICH408I) is issued.
  - No audit records are created.
- VHH1.JBW2000.DBADM in class DSNADM

  Results:

  - No profile is found (return code 4).
  - No failure message (ICH408I) is issued.
  - No audit records are created.
- VHH1.SYSADM in class DSNADM

  Results:

  - Access is granted (return code 0).
  - No failure message (ICH408I) is issued.
  - An audit record is created, which includes the following log string data:
    - The VHH1.BDA0828.EMP.ALTER profile name
    - Input parameters identifying the request from Db2.

Final result

The RACF access control module sends a return code of 0 to Db2.

Example 3: Denying access

The RACF access control module can deny access to Db2 objects.

This example shows how the RACF access control module denies access to a Db2 object (a table). Auditing is activated for all access attempts.

In this example, user ID MIKEJ is trying to alter a table called BDA0828.EMP in database JBW2000.

Setup

- Classification model (&CLASSOPT): 2
- Class name root (&CLASSNMT): DSN
- Class name suffix (&CHAROPT): 1

  This is the default value, but it is not used with supplied classes.
- Db2 subsystem name: VHH1
- Profile:
  - Defined in the MDSNTB class:

    VHH1.BDA0828.EMP.ALTER

    - AUDIT(ALL(READ))
    - UACC(NONE)
    - ID(MIKEJ) ACCESS(NONE)

Profile checking

RACF checks the following resources:

- VHH1.BDA0828.EMP.ALTER in class MDSNTB

  Results:

  - Access is denied (return code 8).
  - No failure message (ICH408I) is issued.
  - No audit records are created.
- VHH1.JBW2000.DBADM in class DSNADM

  Results:

  - No profile is found (return code 4).
  - No failure message (ICH408I) is issued.
  - No audit records are created.
- VHH1.SYSADM in class DSNADM

  Results:

  - No profile is found (return code 4).
  - No failure message (ICH408I) is issued.
  - No audit records are created.
- VHH1.BDA0828.EMP.ALTER in class MDSNTB

  Results:

  - Access is denied (return code 8).
  - Failure message (ICH408I) is issued.
  - An audit record is created, which includes the following log string data:
    - The VHH1.BDA0828.EMP.ALTER profile name
    - Input parameters identifying the request from Db2.

Final result

The RACF access control module sends a return code of 8 to Db2.

Example 4: Deferring to Db2

The RACF access control module can defer to native Db2 authorization checking.

This example shows how the RACF access control module defers to native Db2 authorization checking because the Db2 object (a table) is not protected by RACF.

In this example, user ID MIKEJ is trying to alter a table called BDA0828.EMP in database JBW2000.

Setup

- Classification model (&CLASSOPT): 2
- Class name root (&CLASSNMT): DSN
- Class name suffix (&CHAROPT): 1

  This is the default value, but it is not used with supplied classes.
- Db2 subsystem name: VHH1
- **Profiles:**
  - Defined in the MDSNTB class:

    VHH1.BDASCH1.EMP.ALTER
  - Defined in the DSNADM class:

    VHH1.SYSOPR

    - AUDIT(ALL(READ))

- User ID MIKEJ has SYSOPR authority.

Profile checking

RACF checks the following resources:

1. VHH1.BDA0828.EMP.ALTER in class MDSNTB

   Results:

   - No profile is found (return code 4).
   - No failure message (ICH408I) is issued.
   - No audit records are created.
2. VHH1.JBW2000.DBADM in class DSNADM

   Results:

   - No profile is found (return code 4).
   - No failure message (ICH408I) is issued.
   - No audit records are created.
3. VHH1.SYSADM in class DSNADM

   Results:

   - No profile is found (return code 4).
   - No failure message (ICH408I) is issued.
   - No audit records are created.

Final result

The RACF access control module sends a return code of 4 to Db2.

Example 5: Allowing access (multiple-subsystem scope)

The RACF access control module can grant access to Db2 objects based on a Db2 administrative authority profile.

This example shows how the RACF access control module allows access to a Db2 object (a table) based on a Db2 administrative authority profile. The installation has defined classes MSLH1TB1 and SLH1ADM1. Auditing is activated for all access attempts.

In this example, user ID MIKEJ is trying to alter a table called BDA0828.EMP in database JBW2000.

Setup

- Classification model (&CLASSOPT): 2
- Class name root (&CLASSNMT): SLH1
- Class name suffix (&CHAROPT): 1
- Db2 subsystem name: VHH1
- **Profiles:**
  - Defined in the MSLH1TB1 class:

    VHH1.BDA0828.EMP.ALTER

    - AUDIT(ALL(READ))

    - UACC(NONE)
  - Defined in the SLH1ADM1 class:

VHH1.SYSADM

```
- AUDIT(ALL(READ))
- UACC(NONE)
- ID(MIKEJ) ACCESS(READ)
```

- User ID MIKEJ has SYSADM authority.

Profile checking

RACF checks the following resources:

1. VHH1.BDA0828.EMP.ALTER in class MSLH1TB1

   Results:

   - Access is denied (return code 8).
   - No failure message (ICH408I) is issued.
   - No audit records are created.
2. VHH1.JBW2000.DBADM in class SLH1ADM1

   Results:

   - No profile is found (return code 4).
   - No failure message (ICH408I) is issued.
   - No audit records are created.
3. VHH1.SYSADM in class SLH1ADM1

   Results:

   - Access is granted (return code 0).
   - No failure message (ICH408I) is issued.
   - An audit record is created, which includes the following log string data:
     - The VHH1.BDA0828.EMP.ALTER profile name
     - Input parameters identifying the request from Db2.

Final result

The RACF access control module sends a return code of 0 to Db2.

Example 6: Allowing access (single-subsystem scope)

The RACF access control module can grant access to Db2 objects based on a Db2 administrative authority profile.

This example shows how the RACF access control module allows access to a Db2 object (a table) based on a Db2 administrative authority profile. The installation has defined classes MVHH1TB1 and VHH1ADM1. Auditing is activated for all access attempts.

In this example, user ID MIKEJ is trying to alter a table called BDA0828.EMP in database JBW2000.

Setup

- Classification model (&CLASSOPT): 1
- Class name root (&CLASSNMT): DSN

  This is the default value, but it is not used in single-subsystem scope.
- Class name suffix (&CHAROPT): 1
- Db2 subsystem name: VHH1

- **Profiles:**
  - Defined in the MVHH1TB1 class:

    VHH1.BDA0828.EMP.ALTER

    - AUDIT(ALL(READ))
    - UACC(NONE)
  - Defined in the VHH1ADM1 class:

    SYSADM

    - AUDIT(ALL(READ))
    - UACC(NONE)
    - ID(MIKEJ) ACCESS(READ)
- User ID MIKEJ has SYSADM authority.

Profile checking

RACF checks the following resources:

- BDA0828.EMP.ALTER in class MVHH1TB1

  Results:

  - Access is denied (return code 8).
  - No failure message (ICH408I) is issued.
  - No audit records are created.
- JBW2000.DBADM in class VHH1ADM1

  Results:

  - No profile is found (return code 4).
  - No failure message (ICH408I) is issued.
  - No audit records are created.
- SYSADM in class VHH1ADM1

  Results:

  - Access is granted (return code 0).
  - No failure message (ICH408I) is issued.
  - An audit record is created, which includes the following log string data:
    - The VHH1.BDA0828.EMP.ALTER profile name
    - Input parameters identifying the request from Db2.

Final result

The RACF access control module sends a return code of 0 to Db2.

Chapter 15. RACF authorization checking reference

You can use the RACF access control module to perform RACF authorization checking for several Db2 objects.

This topic includes information about the RACF authorization checking through the RACF access control module for the following Db2 objects:

B
Buffer pools

C
Collections

D
Databases

E
User-defined distinct types

F
User-defined functions

H
Global variables

J
Java archives (JARs)

K
Packages

L
Roles

M
Schemas

N
Trusted contexts

O
Stored procedures

P
Application plans

Q
Sequences

R
Tablespaces

S
Storage groups

T
Tables

U
Systems

V
Views

The sections that follow outline the series of authorization checks that occur in the RACF access control module to determine if the requesting user is authorized to use a particular Db2 privilege against a particular Db2 object type. If any authorization check in the series is successful, the privilege is

granted. For examples of authorization processing in the RACF access control module, see Chapter 14, “Authorization processing examples,” on page 75.

In order to perform authorization checks, the RACF access control module uses the values passed with the following parameters to determine the Db2 object types and privileges:

XAPLTYPE
Db2 object type

XAPLPRIV
Db2 privilege

Restriction: The sections that follow show only the *name* of each Db2 privilege passed with the XAPLPRIV parameter. The RACF access control module uses a numeric XAPLPRIV value. See the Db2 macro DSNXAPRV in *prefix*.SDSNMACS to find the numeric value associated with each Db2 privilege name.

The profile name formats shown in this information are applicable if you are using multiple-subsystem scope (&CLASSOPT 2). If you are using single-subsystem scope (&CLASSOPT 1), the resource name does not include the Db2 subsystem name. If you are using Db2 data sharing, substitute *Db2-group-attachment-name* for *Db2-subsystem* in the profile name formats shown in this appendix.

Note: Having a database privilege on database DSNDB04 is the equivalent of having the privilege on any implicit database. After a privilege is granted, the authorization information is cached for faster re-checking. If the AUTHEXIT_CACHEREFRESH system parameter is specified and RACF commands are issued with generic character ** or * in the resource names, the entire authorization cache for the corresponding class being revoked might be refreshed. In this case, the performance of authorization checking might be impacted until the cache is successfully rebuilt.

When a privilege required by a package is revoked in RACF, the package is not automatically invalidated in Db2. If you want to invalidate the package or make it inoperative, you can use the SQL GRANT statement to grant the revoked privilege and then use the REVOKE statement to revoke it. Or, you can set the AUTHEXIT_CACHEREFRESH system parameter to ALL. See Invalid and inoperative packages (Managing Security) and AUTH EXIT CACHE REFR (AUTHEXIT_CACHEREFRESH subsystem parameter) (Db2 Installation and Migration) for more information.

How to set the level of access

The level of access to Db2 objects, privileges, and administrative authorities is affected by the RACF MLS configuration option.

About this task

When the system is configured with the RACF MLS option not active, access to Db2 objects, privileges or administrative authorities is allowed if the user or group requesting access is in the access list of the RACF profile protecting the object, privilege or authority with at least READ access. If the system is configured with the RACF MLS option active, any operation that performs a write operation (such as UPDATE to a table) must have UPDATE authority (rather than READ).

Note: Use of UPDATE access regardless of the configuration rather than READ in one configuration and UPDATE in another has no effect on access protection and eases administration.

Buffer pool privileges

Resources: Buffer pools

Resource type: B

Db2 privileges

USE

XAPLPRIV value: **USEAUTB**

Privcode 87 (x'57')

The user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.buffer-pool-name.USE	MDSNBP or GDSNBP
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

Collection privileges

Resources: Collections

Resource type: C

Db2 administrative authorities

PACKADM

XAPLPRIV value: **PKADMAUTC**

Privcode 242 (x'F2')

The user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.collection-ID.PACKADM	DSNADM
Db2-subsystem.SYSADM	DSNADM

Db2 privileges

CREATE IN

XAPLPRIV value: **CRTINAUTC**

Privcode 226 (x'E2')

The user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.collection-ID.CREATEIN	MDSNCL or GDSNCL
Db2-subsystem.collection-ID.PACKADM	DSNADM
Db2-subsystem.SYSDBADM	DSNADM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

Database privileges

Resources: Databases

Resource type: D

Note: Having a database privilege on database DSNDB04 is the equivalent of having the privilege on any implicit database.

Db2 administrative authority

DBCTRL

XAPLPRIV value: **DBCTLAUTD**

Privcode 68 (x'44')

The user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.database-name.DBCTRL	DSNADM
Db2-subsystem.database-name.DBADM	DSNADM
Db2-subsystem.SYSDBADM	DSNADM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

Note: If the database was created implicitly, *database-name* must be DSNDB04, not the name of the implicit database.

Db2 privileges

Check Data Utility

XAPLPRIV value: **CHKDAUTD**

Privcode 295 (x'127')

The user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.database-name.STATS	MDSNDB or GDSNDB
Db2-subsystem.database-name.DBMAINT	DSNADM
Db2-subsystem.database-name.DBCTRL	DSNADM
Db2-subsystem.database-name.DBADM	DSNADM
Db2-subsystem.DATAACCESS	DSNADM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

Note: If the database was created implicitly, *database-name* must be DSNDB04, not the name of the implicit database.

CREATETAB

XAPLPRIV value: **CRTTBAUTD**

Privcode 94 (x'5E')

The user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.database-name.CREATETAB	MDSNDB or GDSNDB
Db2-subsystem.database-name.DBMAINT	DSNADM
Db2-subsystem.database-name.DBCTRL	DSNADM
Db2-subsystem.database-name.DBADM	DSNADM
Db2-subsystem.SYSDBADM	DSNADM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

Note: If the database was created implicitly, *database-name* must be DSNDB04, not the name of the implicit database.

CHANGE NAME QUALIFIER

XAPLPRIV value: **QUALAUTD**

Privcode 76 (x'4C')

The user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.database-name.DBCTRL	DSNADM
Db2-subsystem.database-name.DBADM	DSNADM
Db2-subsystem.SYSDBADM	DSNADM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

Note: If the database was created implicitly, *database-name* must be DSNDB04, not the name of the implicit database.

CREATETS

XAPLPRIV value: **CRTTSAUTD**

Privcode 95 (x'5F')

The user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.database-name.CREATETS	MDSNDB or GDSNDB
Db2-subsystem.database-name.DBMAINT	DSNADM
Db2-subsystem.database-name.DBCTRL	DSNADM
Db2-subsystem.database-name.DBADM	DSNADM
Db2-subsystem.SYSDBADM	DSNADM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

Note: If the database was created implicitly, *database-name* must be DSNDB04, not the name of the implicit database.

DISPLAYDB

XAPLPRIV value: **DSPDBAUTD**

Privcode 99 (x'63')

If the database was created implicitly, and the user or the role associated with the user owns the "other object" (XAPLUPRM is equal to XAPLOON when XAPLOOOT indicates an authorization ID, or XAPLUCHK is equal to XAPLOON and XAPLUCKT is equal to XAPLOOT), access is allowed.

If not, the user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.database-name.DISPLAYDB	MDSNDB or GDSNDB
Db2-subsystem.database-name.DBMAINT	DSNADM
Db2-subsystem.database-name.DBCTRL	DSNADM
Db2-subsystem.database-name.DBADM	DSNADM
Db2-subsystem.SYSOPR	DSNADM
Db2-subsystem.DISPLAY	MDSNSM or GDSNSM
Db2-subsystem.SYSDBADM	DSNADM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

Note: If the database was created implicitly, *database-name* must be DSNDB04, not the name of the implicit database.

DROP

XAPLPRIV value: **DROPAUTD**

Privcode 73 (x'49')

The user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.database-name.DROP	MDSNDB or GDSNDB
Db2-subsystem.database-name.DBCTRL	DSNADM
Db2-subsystem.database-name.DBADM	DSNADM
Db2-subsystem.SYSDBADM	DSNADM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

Note: If the database was created implicitly, *database-name* must be DSNDB04, not the name of the implicit database.

MERGECOPY

XAPLPRIV value: **MERGEAUTD**

Privcode 237 (x'ED')

If the database was created implicitly, and the user or the role associated with the user owns the "other object" (XAPLUPRM is equal to XAPLOON when XAPLOOOT indicates an authorization ID, or XAPLUCHK is equal to XAPLOON and XAPLUCKT is equal to XAPLOOT), access is allowed.

If not, the user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.database-name.IMAGCOPY	MDSNDB or GDSNDB
Db2-subsystem.database-name.DBMAINT	DSNADM
Db2-subsystem.database-name.DBCTRL	DSNADM
Db2-subsystem.database-name.DBADM	DSNADM
Db2-subsystem.DATAACCESS	DSNADM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

Note: If the database was created implicitly, *database-name* must be DSNDB04, not the name of the implicit database.

IMAGCOPY, MODIFY RECOVERY, QUIESCE

XAPLPRIV values: **IMCOPAUTD, MODAUTD, QUIESAUTD**

Privcode 74 (x'4A'), 238 (x'EE'), 239 (x'EF')

If the database was created implicitly, and the user or the role associated with the user owns the "other object" (XAPLUPRM is equal to XAPLOON when XAPLOOOT indicates an authorization ID, or XAPLUCHK is equal to XAPLOON and XAPLUCKT is equal to XAPLOOT), access is allowed.

If not, the user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.database-name.IMAGCOPY	MDSNDB or GDSNDB
Db2-subsystem.database-name.DBMAINT	DSNADM
Db2-subsystem.database-name.DBCTRL	DSNADM
Db2-subsystem.database-name.DBADM	DSNADM
Db2-subsystem.SYSDBADM	DSNADM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

Note: If the database was created implicitly, *database-name* must be DSNDB04, not the name of the implicit database.

RECOVERDB, REPORT

XAPLPRIV values: **RECDBAUTD, REPRTAUTD**

Privcode 89 (x'59'), 240 (x'F0')

If the database was created implicitly, and the user or the role associated with the user owns the "other object" (XAPLUPRM is equal to XAPLOON when XAPLOOOT indicates an authorization ID, or XAPLUCHK is equal to XAPLOON and XAPLUCKT is equal to XAPLOOT), access is allowed.

If not, the user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.database-name.RECOVERDB	MDSNDB or GDSNDB
Db2-subsystem.database-name.DBCTRL	DSNADM
Db2-subsystem.database-name.DBADM	DSNADM
Db2-subsystem.SYDBADM	DSNADM
Db2-subsystem.DATAACCESS	DSNADM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

Note: If the database was created implicitly, *database-name* must be DSNDB04, not the name of the implicit database.

REORG

XAPLPRIV value: **REORGAUTD**

Privcode 77 (x'4D')

If the database was created implicitly, and the user or the role associated with the user owns the "other object" (XAPLUPRM is equal to XAPLOON when XAPLOOOT indicates an authorization ID, or XAPLUCHK is equal to XAPLOON and XAPLUCKT is equal to XAPLOOT), access is allowed.

If not, the user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.database-name.REORG	MDSNDB or GDSNDB
Db2-subsystem.database-name.DBCTRL	DSNADM
Db2-subsystem.database-name.DBADM	DSNADM
Db2-subsystem.DATAACCESS	DSNADM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

Note: If the database was created implicitly, *database-name* must be DSNDB04, not the name of the implicit database.

REPAIR

XAPLPRIV values: **REPARAUTD**

Privcode 78 (x'4E')

If the database was created implicitly, and the user or the role associated with the user owns the "other object" (XAPLUPRM is equal to XAPLOON when XAPLOOOT indicates an authorization ID, or XAPLUCHK is equal to XAPLOON and XAPLUCKT is equal to XAPLOOT), access is allowed.

If not, the user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.database-name.REPAIR	MDSNDB or GDSNDB
Db2-subsystem.database-name.DBCTRL	DSNADM
Db2-subsystem.database-name.DBADM	DSNADM

One of these resources:	In class:
Db2-subsystem.DATAACCESS	DSNADM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

Note: If the database was created implicitly, *database-name* must be DSNDB04, not the name of the implicit database.

RUN REPAIR UTILITY

XAPLPRIV values: **DIAGAUTD**

Privcode 236 (x'EC')

If the database was created implicitly, and the user or the role associated with the user owns the "other object" (XAPLUPRM is equal to XAPLOON when XAPLOOOT indicates an authorization ID, or XAPLUCHK is equal to XAPLOON and XAPLUCKT is equal to XAPLOOT), access is allowed.

If not, the user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.database-name.REPAIR	MDSNDB or GDSNDB
Db2-subsystem.database-name.DBCTRL	DSNADM
Db2-subsystem.database-name.DBADM	DSNADM
Db2-subsystem.SQLADM	MDSNSM or GDSNSM
Db2-subsystem.SYSDBADM	DSNADM
Db2-subsystem.DATAACCESS	DSNADM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

Note: If the database was created implicitly, *database-name* must be DSNDB04, not the name of the implicit database.

REPAIR DBD

XAPLPRIV value: **RDBDAUTD**

Privcode 241 (x'F1')

The user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

RUN CHECK INDEX/LOB UTILITY

XAPLPRIV values: **CHECKAUTD**

Privcode 19 (x'13')

If the database was created implicitly, and the user or the role associated with the user owns the "other object" (XAPLUPRM is equal to XAPLOON when XAPLOOOT indicates an authorization ID, or XAPLUCHK is equal to XAPLOON and XAPLUCKT is equal to XAPLOOT), access is allowed.

If not, the user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.database-name.STATS	MDSNDB or GDSNDB
Db2-subsystem.database-name.DBMAINT	DSNADM
Db2-subsystem.database-name.DBCTRL	DSNADM
Db2-subsystem.database-name.DBADM	DSNADM
Db2-subsystem.SYSDBADM	DSNADM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

Note: If the database was created implicitly, *database-name* must be DSNDB04, not the name of the implicit database.

STATS

XAPLPRIV values: **STATSAUTD**

Privcode 82 (x'52')

If the database was created implicitly, and the user or the role associated with the user owns the "other object" (XAPLUPRM is equal to XAPLOON when XAPLOOOT indicates an authorization ID, or XAPLUCHK is equal to XAPLOON and XAPLUCKT is equal to XAPLOOT), access is allowed.

If not, the user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.database-name.STATS	MDSNDB or GDSNDB
Db2-subsystem.database-name.DBMAINT	DSNADM
Db2-subsystem.database-name.DBCTRL	DSNADM
Db2-subsystem.database-name.DBADM	DSNADM
Db2-subsystem.SQLADM	MDSNSM or GDSNSM
Db2-subsystem.SYSDBADM	DSNADM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

Note: If the database was created implicitly, *database-name* must be DSNDB04, not the name of the implicit database.

STARTDB

XAPLPRIV value: **STARTAUTD**

Privcode 79 (x'4F')

If the database was created implicitly, and the user or the role associated with the user owns the "other object" (XAPLUPRM is equal to XAPLOON when XAPLOOOT indicates an authorization ID, or XAPLUCHK is equal to XAPLOON and XAPLUCKT is equal to XAPLOOT), access is allowed.

If not, the user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.database-name.STARTDB	MDSNDB or GDSNDB
Db2-subsystem.database-name.DBMAINT	DSNADM
Db2-subsystem.database-name.DBCTRL	DSNADM
Db2-subsystem.database-name.DBADM	DSNADM
Db2-subsystem.SYSDBADM	DSNADM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

Note: If the database was created implicitly, *database-name* must be DSNDB04, not the name of the implicit database.

STOPDB

XAPLPRIV value: **STOPAUTD**

Privcode 83 (x'53')

If the database was created implicitly, and the user or the role associated with the user owns the "other object" (XAPLUPRM is equal to XAPLOON when XAPLOOOT indicates an authorization ID, or XAPLUCHK is equal to XAPLOON and XAPLUCKT is equal to XAPLOOT), access is allowed.

If not, the user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.database-name.STOPDB	MDSNDB or GDSNDB
Db2-subsystem.database-name.DBMAINT	DSNADM
Db2-subsystem.database-name.DBCTRL	DSNADM
Db2-subsystem.database-name.DBADM	DSNADM
Db2-subsystem.SYSDBADM	DSNADM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

Note: If the database was created implicitly, *database-name* must be DSNDB04, not the name of the implicit database.

TERM UTILITY

XAPLPRIV value: **TERMAUTD**

Privcode 109 (x'6D')

The user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.SYSDBADM	DSNADM
Db2-subsystem.DATAACCESS	DSNADM
Db2-subsystem.SYSOPR	DSNADM

One of these resources:	In class:
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

TERM UTILITY ON DATABASE

XAPLPRIV value: **TERMDAUTD**

Privcode 58 (x'3A')

If the database was created implicitly, and the user or the role associated with the user owns the "other object" (XAPLUPRM is equal to XAPLOON when XAPLOOOT indicates an authorization ID, or XAPLUCHK is equal to XAPLOON and XAPLUCKT is equal to XAPLOOT), access is allowed.

If not, the user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.database-name.DBMAINT	DSNADM
Db2-subsystem.database-name.DBCTRL	DSNADM
Db2-subsystem.database-name.DBADM	DSNADM

Note: If the database was created implicitly, *database-name* must be DSNDB04, not the name of the implicit database.

Global variable privileges

Resources: Global variables

Resource type: H

Db2 privileges

READ

XAPLPRIV value: **READAUTH**

Privcode 291(x'123')

Does the user or the role associated with the user own the variable?

If so, XAPLUPRM must match the owner name that is passed from Db2 by the XAPLOWNR parameter when XAPLONRT indicates that an authorization ID XAPLUCHK must match XAPLOWNR, and an authorization ID XAPLUCKT must match XAPLONRT.

If XAPLACAC is on (XAPLFLG2 bit 5 is '1'B) and XAPLUCHK is an authorization ID, suppress the ownership check for XAPLUCHK.

If not, the user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.schema-name.variable-name.READ	MDSNGV or GDSNGV
Db2-subsystem.DATAACCESS	DSNADM
Db2-subsystem.SYSADM	DSNADM

WRITE

XAPLPRIV value: **WRITEAUTH**

Privcode 292 (x'124')

Does the user or the role associated with the user own the variable?

If so, XAPLUPRM must match the owner name that is passed from Db2 by the XAPLOWNR parameter when XAPLONRT indicates that an authorization ID XAPLUCHK must match XAPLOWNR and an authorization ID XAPLUCKT must match XAPLONRT.

If XAPLACAC is on (XAPLFLG2 bit 5 is '1'B) and XAPLUCHK is an an authorization ID, suppress the ownership check for XAPLUCHK.

If not, the user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.schema-name.variable-name.WRITE	MDSNGV or GDSNGV
Db2-subsystem.DATAACCESS	DSNADM
Db2-subsystem.SYSADM	DSNADM

Java archive (JAR) privileges

Resources: Java archives (JARs)

Resource type: J

Db2 privileges

USAGE

XAPLPRIV value: **USAGEAUTJ**

Privcode 263 (x'107')

Does the user or the role associated with the user own the Java archive (JAR)?

If so, XAPLUPRM must match the owner name passed from Db2 by the XAPLOWNR parameter when XAPLONRT indicates an authorization ID, or XAPLUCHK must match XAPLOWNR and XAPLUCKT must match XAPLONRT.

If not, the user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.schema-name.JAR-name.USAGE	MDSNJR or GDSNJR
Db2-subsystem.DATAACCESS	DSNADM
Db2-subsystem.SYSADM	DSNADM

Package privileges

Resources: Packages

Resource type: K

Db2 privileges

BIND

XAPLPRIV value: **BINDAUTK**

Privcode 65 (x'41')

Does the user or the role associated with the user own the package?

If so, XAPLUPRM must match the owner name passed from Db2 by the XAPLOWNR parameter when XAPLONRT indicates an authorization ID, or XAPLUCHK must match XAPLOWNR and XAPLUCKT must match XAPLONRT.

If not, the user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.collection-ID.package-ID.BIND	MDSNPK or GDSNPK
Db2-subsystem.owner.BINDAGENT	MDSNSM or GDSNSM
Db2-subsystem.collection-ID.PACKADM	DSNADM
Db2-subsystem.SYSDBADM	DSNADM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

COMMENT ON

XAPLPRIV value: **COMNTAUTK**

Privcode 97 (x'61')

Does the user or the role associated with the user own the package?

If so, XAPLUPRM must match the owner name passed from Db2 by the XAPLOWNR parameter when XAPLONRT indicates an authorization ID, or XAPLUCHK must match XAPLOWNR and XAPLUCKT must match XAPLONRT.

If not, the user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.owner.BINDAGENT	MDSNSM or GDSNSM
Db2-subsystem.collection-ID.PACKADM	DSNADM
Db2-subsystem.SYSDBADM	DSNADM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

COPY

XAPLPRIV value: **COPYAUTK**

Privcode 225 (x'E1')

Does the user or the role associated with the user own the package?

If so, XAPLUPRM must match the owner name passed from Db2 by the XAPLOWNR parameter when XAPLONRT indicates an authorization ID, or XAPLUCHK must match XAPLOWNR and XAPLUCKT must match XAPLONRT.

If not, the user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.collection-ID.package-ID.COPY	MDSNPK or GDSNPK
Db2-subsystem.owner.BINDAGENT	MDSNSM or GDSNSM

One of these resources:	In class:
Db2-subsystem.collection-ID.PACKADM	DSNADM
Db2-subsystem.SYSDBADM	DSNADM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

DROP

XAPLPRIV value: **DROPAUTK**

Privcode 73 (x'49')

The user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.owner.BINDAGENT	MDSNSM or GDSNSM
Db2-subsystem.collection-ID.PACKADM	DSNADM
Db2-subsystem.SYSDBADM	DSNADM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

EXECUTE

XAPLPRIV value: **CHKEXECK**

Privcode 64 (x'40')

The user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.collection-ID.package-ID.EXECUTE	MDSNPK or GDSNPK
Db2-subsystem.collection-ID.PACKADM	DSNADM
Db2-subsystem.SQLADM This check is only done for system defined packages.	MDSNSM or GDSNSM
Db2-subsystem.SYSDBADM This check is only done for system defined packages.	DSNADM
Db2-subsystem.DATAACCESS	DSNADM
Db2-subsystem.SYSADM	DSNADM

All package privileges (PACKADM or SYSADM)

XAPLPRIV value: **ALLPKAUTK**

Privcode 228 (x'E4')

There are no authorization checks (return code 4).

All package privileges

XAPLPRIV value: **SUBPKAUTK**

Privcode 229 (x'E5')

The user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.collection-ID.PACKADM The user has authority to *collection-ID*.	DSNADM
Db2-subsystem.ACCESSCTRL	DSNADM
Db2-subsystem.SYSCTRL Bypass if SEPARATE_SECURITY= Yes	DSNADM
Db2-subsystem.SYSADM Bypass if SEPARATE_SECURITY= Yes	DSNADM
Db2-subsystem.SECADM	DSNADM

Plan privileges

Resources: Application plans

Resource type: P

Db2 privileges

BIND

XAPLPRIV value: **BINDAUTP**

Privcode 65 (x'41')

Does the user or the role associated with the user own the plan?

If so, XAPLUPRM must match the owner name passed from Db2 by the XAPLOWNR parameter when XAPLONRT indicates an authorization ID, or XAPLUCHK must match XAPLOWNR and XAPLUCKT must match XAPLONRT.

If not, the user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.plan-name.BIND	MDSNPN or GDSNPN
Db2-subsystem.owner.BINDAGENT	MDSNSM or GDSNSM
Db2-subsystem.SYSDBADM	DSNADM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

COMMENT ON

XAPLPRIV value: **COMNTAUTP**

Privcode 97 (x'61')

Does the user or the role associated with the user own the plan?

If so, XAPLUPRM must match the owner name passed from Db2 by the XAPLOWNR parameter when XAPLONRT indicates an authorization ID, or XAPLUCHK must match XAPLOWNR and XAPLUCKT must match XAPLONRT.

If not, the user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.owner.BINDAGENT	MDSNSM or GDSNSM
Db2-subsystem.SYSDBADM	DSNADM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

EXECUTE

XAPLPRIV value: **CHKEXECP**

Privcode 64 (x'40')

The user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.plan-name.EXECUTE	MDSNPN or GDSNPN
Db2-subsystem.DATAACCESS	DSNADM
Db2-subsystem.SYSADM	DSNADM

Role privileges

Resources: Roles

Resource type: L

Db2 privileges

COMMENT ON ROLE

XAPLPRIV value: **COMNTAUTL**

Privcode 97 (x'61')

Does the user or the role associated with the user own the role?

If so, XAPLUPRM must match the owner name passed from Db2 by the XAPLOWNR parameter when XAPLONRT indicates an authorization ID, or XAPLUCHK must match XAPLOWNR and XAPLUCKT must match XAPLONRT.

If not, the user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.SYSCTRL Bypass if Separate Security = Yes	DSNADM
Db2-subsystem.SYSADM Bypass if Separate Security = Yes	DSNADM
Db2-subsystem.SECADM	DSNADM

CREATE ROLE

XAPLPRIV value: **CREATAUTL**

Privcode 271 (x'10F')

The user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.SYSCTRL Bypass if Separate Security = Yes	DSNADM
Db2-subsystem.SYSADM Bypass if Separate Security = Yes	DSNADM
Db2-subsystem.SECADM	DSNADM

DROP ROLE

XAPLPRIV value: **DROPAUTL**

Privcode 73 (x'49')

Does the user or the role associated with the user own the role?

If so, XAPLUPRM must match the owner name passed from Db2 by the XAPLOWNR parameter when XAPLONRT indicates an authorization ID, or XAPLUCHK must match XAPLOWNR and XAPLUCKT must match XAPLONRT.

If not, the user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.SYSCTRL Bypass if Separate Security = Yes	DSNADM
Db2-subsystem.SYSADM Bypass if Separate Security = Yes	DSNADM
Db2-subsystem.SECADM	DSNADM

Schema privileges

Resources: Schemas

Resource type: M

Db2 privileges

ALTERIN

XAPLPRIV value: **ALTINAUTM**

Privcode 252 (x'FC')

Does the user match the schema name?

If so, XAPLUPRM or XAPLUCHK must match the schema name passed from Db2 by the XAPLOWNQ parameter.

If not, does the user or the role associated with the user own the object?

If so, XAPLUPRM must match the owner name of the object being altered passed from Db2 by the XAPLOWNR parameter when XAPLONRT indicates an authorization ID, or XAPLUCHK must match XAPLOWNR and XAPLUCKT must match XAPLONRT.

If not, the user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.schema-name.object-name.ALTERIN	MDSNSC or GDSNSC
Db2-subsystem.SYSDBADM	DSNADM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

CHANGE NAME QUALIFIER

XAPLPRIV value: **QUALAUTM**

Privcode 76 (x'4C')

The user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.SYSDBADM	DSNADM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

Note: No RACF audit record or ICH408I message is generated for a failure related to this privilege. RACF will audit successes, if specified.

COMMENT ON

XAPLPRIV value: **COMNTAUTM**

Privcode 97 (x'61')

Does the user match the schema name?

If so, XAPLUPRM or XAPLUCHK must match the schema name passed from Db2 by the XAPLOWNQ parameter.

If not, does the user or the role associated with the user own the object?

If so, XAPLUPRM must match the owner name passed from Db2 by the XAPLOWNR parameter when XAPLONRT indicates an authorization ID, or XAPLUCHK must match XAPLOWNR and XAPLUCKT must match XAPLONRT.

If not, the user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.schema-name.object-name.ALTERIN	MDSNSC or GDSNSC
Db2-subsystem.SYSDBADM	DSNADM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

CREATEIN

XAPLPRIV value: **CREINAUTM**

Privcode 261 (x'105')

Does the user match the schema name?

If so, XAPLUPRM or XAPLUCHK must match the schema name passed from Db2 by the XAPLOBJN parameter.

If not, the user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.schema-name.CREATEIN	MDSNSC or GDSNSC
Db2-subsystem.SYSDBADM	DSNADM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

DROPIN

XAPLPRIV value: **DRPINAUTM**

Privcode 262 (x'106')

Does the user match the schema name?

If so, XAPLUPRM or XAPLUCHK must match the schema name passed from Db2 by the XAPLOWNQ parameter.

If not, does the user own the object?

If so, XAPLUPRM or XAPLUCHK must match the owner name passed from Db2 by the XAPLOWNR parameter.

If not, the user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.schema-name.object-name.DROPIN	MDSNSC or GDSNSC
Db2-subsystem.SYSDBADM	DSNADM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

Sequence privileges

Resources: Sequences

Resource type: Q

Db2 privileges

ALTER

XAPLPRIV value: **ALTERAUTQ**

Privcode 61 (x'3D')

Does the user match the schema name?

If so, XAPLUPRM or XAPLUCHK must match the schema name passed from Db2 by the XAPLOWNQ parameter.

If not, does the user or the role associated with the user own the sequence?

If so, XAPLUPRM must match the owner name passed from Db2 by the XAPLOWNR parameter when XAPLONRT indicates an authorization ID, or XAPLUCHK must match XAPLOWNR and XAPLUCKT must match XAPLONRT.

If not, the user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.schema-name.object-name.ALTERIN	MDSNSC or GDSNSC
Db2-subsystem.schema-name.sequence-name.ALTER	MDSNSQ or GDSNSQ
Db2-subsystem.SYSDBADM	DSNADM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

COMMENT ON

XAPLPRIV value: **COMNTAUTQ**

Privcode 97 (x'61')

Does the user match the schema name?

If so, XAPLUPRM or XAPLUCHK must match the schema name passed from Db2 by the XAPLOWNQ parameter.

If not, does the user or the role associated with the user own the sequence?

If so, XAPLUPRM must match the owner name passed from Db2 by the XAPLOWNR parameter when XAPLONRT indicates an authorization ID, or XAPLUCHK must match XAPLOWNR and XAPLUCKT must match XAPLONRT.

If not, the user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.schema-name.object-name.ALTERIN	MDSNSC or GDSNSC
Db2-subsystem.schema-name.sequence-name.ALTER	MDSNSQ or GDSNSQ
Db2-subsystem.SYSDBADM	DSNADM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

USAGE

XAPLPRIV value: **USAGEAUTQ**

Privcode 263 (x'107')

Does the user or the role associated with the user own the sequence?

If so, XAPLUPRM must match the owner name passed from Db2 by the XAPLOWNR parameter when XAPLONRT indicates an authorization ID, or XAPLUCHK must match XAPLOWNR and XAPLUCKT must match XAPLONRT.

If XAPLACAC is enabled (XAPLFLG2 bit 5 is '1'B) and XAPLUCHK is an authid, suppress the ownership check for XAPLUCHK.

If not, the user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.schema-name.sequence-name.USAGE	MDSNSQ or GDSNSQ
Db2-subsystem.DATACCESS	DSNADM
Db2-subsystem.SYSADM	DSNADM

Storage group privileges

Resources: Storage groups

Resource type: S

Db2 privileges

DROP, ALTER

XAPLPRIV values: **DROPAUTS, ALTERAUTS**

Privcode 73 (x'49'), 61 (x'3D')

The user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

USE

XAPLPRIV value: **USEAUTS**

Privcode 87 (x'57')

The user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.storage-groupname.USE	MDSNSG or GDSNSG
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

Stored procedure privileges

Resources: Stored procedures

Resource type: O

Db2 privileges

DISPLAY

XAPLPRIV value: **DISPAUTO**

Privcode 267 (x'10B')

Does the user match the schema name?

If so, XAPLUPRM or XAPLUCHK must match the schema name passed from Db2 by the XAPLOWNQ parameter.

If not, does the user or the role associated with the user own the stored procedure?

If so, XAPLUPRM must match the owner name passed from Db2 by the XAPLOWNR parameter when XAPLONRT indicates an authorization ID, or XAPLUCHK must match XAPLOWNR and XAPLUCKT must match XAPLONRT.

If not, the user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.schema-name.procedure-name.DISPLAY	MDSNSP or GDSNSP
Db2-subsystem.SYSDBADM	DSNADM
Db2-subsystem.SYSOPR	DSNADM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

EXECUTE

XAPLPRIV value: **CHKEXECO**

Privcode 64 (x'40')

Does the user or the role associated with the user own the stored procedure?

If so, XAPLUPRM must match the owner name passed from Db2 by the XAPLOWNR parameter when XAPLONRT indicates an authorization ID, or XAPLUCHK must match XAPLOWNR and XAPLUCKT must match XAPLONRT.

If XAPLACAC is enabled (XAPLFLG2 bit 5 is '1'B) and XAPLUCHK is an authorization ID, suppress the ownership check for XAPLUCHK.

If not, the user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.schema-name.procedure-name.EXECUTE	MDSNSP or GDSNSP
Db2-subsystem.SQLADM This check is performed only for system defined packages.	MDSNSM or GDSNSM
Db2-subsystem.SYSDBADM This check is performed only for system defined packages.	DSNADM
Db2-subsystem.DATAACCESS	DSNADM
Db2-subsystem.SYSADM	DSNADM

START

XAPLPRIV value: **STRTAUTO**

Privcode 265 (x'109')

Does the user match the schema name?

If so, XAPLUPRM or XAPLUCHK must match the schema name passed from Db2 by the XAPLOWNQ parameter.

If not, does the user or the role associated with the user own the stored procedure?

If so, XAPLUPRM must match the owner name passed from Db2 by the XAPLOWNR parameter when XAPLONRT indicates an authorization ID, or XAPLUCHK must match XAPLOWNR and XAPLUCKT must match XAPLONRT.

If not, the user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.SYSDBADM	DSNADM
Db2-subsystem.SYSOPR	DSNADM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

STOP

XAPLPRIV value: **STPAUTO**

Privcode 266 (x'10A')

Does the user match the schema name?

If so, XAPLUPRM or XAPLUCHK must match the schema name passed from Db2 by the XAPLOWNQ parameter.

If not, does the user or the role associated with the user own the stored procedure?

If so, XAPLUPRM must match the owner name passed from Db2 by the XAPLOWNR parameter when XAPLONRT indicates an authorization ID, or XAPLUCHK must match XAPLOWNR and XAPLUCKT must match XAPLONRT.

If not, the user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.SYSDBADM	DSNADM
Db2-subsystem.SYSOPR	DSNADM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

System privileges

Resources: Systems

Resource type: U

Db2 administrative authorities

ACCESSCTRL

XAPLPRIV value: **ACNTLAUTU**

Privcode 289 (x'121')

The user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.ACCESSCTRL	DSNADM

One of these resources:	In class:
Db2-subsystem.SYSCTRL Bypass if SEPARATE_SECURITY= Yes	DSNADM
Db2-subsystem.SYSADM Bypass if SEPARATE_SECURITY= Yes	DSNADM
Db2-subsystem.SECADM	DSNADM

SQLADM

XAPLPRIV value: **SQLAAUTHU**

Privcode 290 (x'122')

The user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.SQLADM	MDSNSM or GDSNSM
Db2-subsystem.SYSDBADM	DSNADM
Db2-subsystem.SYSADM	DSNADM

SECADM

XAPLPRIV value: **SECAAUTHU**

Privcode 284 (x'11C')

The user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.SYSADM Bypass if Separate Security = Yes	DSNADM
Db2-subsystem.SECADM	DSNADM

SYSADM

XAPLPRIV value: **SYSAAUTHU**

Privcode 85 (x'55')

The user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.SYSADM	DSNADM

SYSCTRL

XAPLPRIV value: **SYSCAUTHU**

Privcode 224 (x'E0')

The user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

Note: Having a database privilege on database DSNDB04 is the equivalent of having the privilege on any implicit database.

SYSDBADM

XAPLPRIV value: **DB2AAUTHU**

Privcode 287 (x'11F')

The user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.SYSDBADM	DSNADM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

Note: Db2 turns on bit 7 of the XAPLFLG1 field for a user table that includes user defined data type or user defined function. If this bit is on, the RACF® access control module bypasses checking for the SYSCTRL authority.

SYSOPR

XAPLPRIV value: **SOSEAUTHU**

Privcode 296 (x'128')

The user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.SYSOPR	DSNADM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM
Db2-subsystem.SECADM	DSNADM

Db2 privileges

ALTER BUFFERPOOL

XAPLPRIV value: **CHKALTBPU**

Privcode 113 (x'71')

The user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.SYSOPR	DSNADM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

BINDADD

XAPLPRIV value: **BINDAAUTU**

Privcode 88 (x'58')

The user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.BINDADD	MDSNSM or GDSNSM
Db2-subsystem.SYSDBADM	DSNADM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

BINDAGENT

XAPLPRIV value: **BNDAGAUTU**

Privcode 227 (x'E3')

The user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.owner.BINDAGENT	MDSNSM or GDSNSM
Db2-subsystem.SYSDBADM	DSNADM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

CANCEL DDF THREAD, START | STOP DDF

XAPLPRIV values: **CHKDDFU, CHKDDFU, CHKDDFU**

Privcode 21 (x'15'), 21 (x'15'), 21 (x'15')

The user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.SYSOPR	DSNADM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

START | STOP RLIMIT

XAPLPRIV values: **CHKSTARTU, CHKSTOPU**

Privcode 12 (x'C'), 13 (x'D')

The user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.SYSOPR	DSNADM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

DISPLAY RLIMIT

XAPLPRIV values: **CHKDSPLU**

Privcode 14 (x'E')

The user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.SYSDBADM	DSNADM
Db2-subsystem.SYSOPR	DSNADM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

CREATEALIAS

XAPLPRIV value: **CRTALAUTU**

Privcode 15 (x'F')

The user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.CREATEALIAS	MDSNSM or GDSNSM
Db2-subsystem.SYSDBADM	DSNADM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM
Note: DBADM and DBCTRL authorities can be used to allow a user to create aliases. See "CREATE ALIAS privilege" on page 47 for more information.	
Db2-subsystem.database-name.DBCTRL	DSNADM
Db2-subsystem.database-name.DBADM	DSNADM

CREATEDBA

XAPLPRIV value: **CRTDBAUTU**

Privcode 66 (x'42')

The user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.CREATEDBA	MDSNSM or GDSNSM
Db2-subsystem.CREATEDBC	MDSNSM or GDSNSM
Db2-subsystem.SYSDBADM	DSNADM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

CREATESG

XAPLPRIV value: **CRTSGAUTU**

Privcode 67 (x'43')

The user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.CREATESG	MDSNSM or GDSNSM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

CREATETMTAB

XAPLPRIV value: **CRTTMAUTU**

Privcode 248 (x'F8')

The user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.CREATETMTAB	MDSNSM or GDSNSM
Db2-subsystem.CREATETAB	MDSNDB or GDSNDB
Db2-subsystem.SYSDBADM	DSNADM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

Create Secure Object

XAPLPRIV value: **CRTSOAUTU**

Privcode 285 (x'11D')

The user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.CREATESECUREOBJECT	MDSNSM or GDSNSM
Db2-subsystem.SYSADM	DSNADM
Bypass if Separate Security = yes	
Db2-subsystem.SECADM	DSNADM

DEBUGSESSION

XAPLPRIV value: **DEBUGAUTU**

Privcode 282 (x'11A')

The user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.DEBUGSESSION	MDSNSM or GDSNSM
Db2-subsystem.DATAACCESS	DSNADM
Db2-subsystem.SYSADM	DSNADM

DISPLAY, DISPLAY BUFFERPOOL

XAPLPRIV values: **CHKDISPLU, CHKDSPBPU**

Privcode 62 (x'3E'), 112 (x'70')

The user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.DISPLAY	MDSNSM or GDSNSM
Db2-subsystem.SYSDBADM	DSNADM
Db2-subsystem.SYSOPR	DSNADM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

DISPLAY ARCHIVE

XAPLPRIV value: **DARCHAUTU**

Privcode 244 (x'F4')

The user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.DISPLAY	MDSNSM or GDSNSM
Db2-subsystem.ARCHIVE	MDSNSM or GDSNSM
Db2-subsystem.SYSDBADM	DSNADM
Db2-subsystem.SYSOPR	DSNADM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

DISPLAY PROFILE

XAPLPRIV value: **CHKDSPPU**

Privcode 9 (x'9')

The user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.SQLADM	MDSNSM or GDSNSM
Db2-subsystem.SYSDBADM	DSNADM
Db2-subsystem.SYSOPR	DSNADM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

Explain

XAPLPRIV value: **EXPLNAUTU**

Privcode 286 (x'11E')

The user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.EXPLAIN	MDSNSM or GDSNSM
Db2-subsystem.SQLADM	MDSNSM or GDSNSM
Db2-subsystem.SYSDBADM	DSNADM
Db2-subsystem.SYSADM	DSNADM

MONITOR1

XAPLPRIV value: **MON1AUTU**

Privcode 16 (x'10')

The user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.MONITOR1	MDSNSM or GDSNSM
Db2-subsystem.MONITOR2	MDSNSM or GDSNSM
Db2-subsystem.SQLADM	MDSNSM or GDSNSM
Db2-subsystem.SYSDBADM	DSNADM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

MONITOR2

XAPLPRIV value: **MON2AUTU**

Privcode 17 (x'11')

The user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.MONITOR2	MDSNSM or GDSNSM
Db2-subsystem.SQLADM	MDSNSM or GDSNSM
Db2-subsystem.SYSDBADM	DSNADM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

Query Tuning

XAPLPRIV value: **QRYTAUTU**

Privcode 294 (x'126')

The user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.SQLADM	MDSNSM or GDSNSM
Db2-subsystem.SYSDBADM	DSNADM
Db2-subsystem.SYSOPR	DSNADM

One of these resources:	In class:
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

RECOVER BSDS

XAPLPRIV value: **CHKBSDSU**

Privcode 93 (x'5D')

The user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.BSDS	MDSNSM or GDSNSM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

RECOVER INDOUBT

XAPLPRIV value: **CHKRECOVU**

Privcode 72 (x'48')

The user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.RECOVER	MDSNSM or GDSNSM
Db2-subsystem.SYSDBADM	DSNADM
Db2-subsystem.SYSOPR	DSNADM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

SET ARCHIVE

XAPLPRIV value: **SARCHAUTU**

Privcode 243 (x'F3')

The user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.ARCHIVE	MDSNSM or GDSNSM
Db2-subsystem.SYSOPR	DSNADM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

START PROFILE

XAPLPRIV value: **CHKSTRTPU**

Privcode 10 (x'A')

The user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.SQLADM	MDSNSM or GDSNSM
Db2-subsystem.SYSDBADM	DSNADM
Db2-subsystem.SYSOPR	DSNADM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

STOP PROFILE

XAPLPRIV value: **CHKSTOPPU**

Privcode 11 (x'B')

The user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.SQLADM	MDSNSM or GDSNSM
Db2-subsystem.SYSDBADM	DSNADM
Db2-subsystem.SYSOPR	DSNADM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

STOPALL

XAPLPRIV value: **CHKSUBSYU**

Privcode 80 (x'50')

The user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.STOPALL	MDSNSM or GDSNSM
Db2-subsystem.SYSOPR	DSNADM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

STOSPACE UTILITY

XAPLPRIV value: **STOAUTU**

Privcode 107 (x'6B')

The user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.STOSPACE	MDSNSM or GDSNSM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

START | STOP | MODIFY TRACE

XAPLPRIV value: **CHKTRACEU**

Privcode 84 (x'54')

The user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.TRACE	MDSNSM or GDSNSM
Db2-subsystem.SQLADM	MDSNSM or GDSNSM
Db2-subsystem.SYSDBADM	DSNADM
Db2-subsystem.SYSOPR	DSNADM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM
Db2-subsystem.SECADM	DSNADM

USE ARCHIVE LOG

XAPLPRIV value: **ARCHAUTU**

Privcode 231 (x'E7')

The user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.ARCHIVE	MDSNSM or GDSNSM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

Table privileges

Resources: Tables

Resource type: T

Note about SYSCTRL

The SYSCTRL administrative authority does not apply to user tables. Db2 turns on bit 7 of the XAPLFLG1 field for a user table. If this bit is on, the RACF access control module bypasses checking for the SYSCTRL authority. This allows RACF processing to model Db2 processing.

Note: Having a database privilege on database DSNDB04 is the equivalent of having the privilege on any implicit database.

Db2 privileges

ALTER

XAPLPRIV value: **ALTERAUTT**

Privcode 61 (x'3D')

Does the user or the role associated with the user own the table?

If so, XAPLUPRM must match the owner name passed from Db2 by the XAPLOWNR parameter when XAPLONRT indicates an authorization ID, or XAPLUCHK must match XAPLOWNR and XAPLUCKT must match XAPLONRT.

If not, the user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.table-qualifier.table-name.ALTER	MDSNTB or GDSNTB
Db2-subsystem.database-name.DBADM	DSNADM
Db2-subsystem.SYSDBADM	DSNADM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

ALTER INDEX, DROP INDEX

XAPLPRIV values: **ALTIXAUTT, DRPIXAUTT**

Privcode 103 (x'67'), 105 (x'69')

Does the user or the role associated with the user own the index?

If so, XAPLUPRM must match the owner name passed from Db2 by the XAPLOWNR parameter when XAPLONRT indicates an authorization ID, or XAPLUCHK must match XAPLOWNR and XAPLUCKT must match XAPLONRT.

If not, the user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.database-name.DBADM	DSNADM
Db2-subsystem.SYSDBADM	DSNADM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

CHANGE NAME QUALIFIER

XAPLPRIV value: **QUALAUTT**

Privcode 76 (x'4C')

The user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.database-name.DBCTRL	DSNADM
Db2-subsystem.database-name.DBADM	DSNADM
Db2-subsystem.SYSDBADM	DSNADM
Db2-subsystem.SYSCTRL This check is bypassed for user tables.	DSNADM
Db2-subsystem.SYSADM	DSNADM

COMMENT ON, COMMENT ON INDEX, DROP

XAPLPRIV values: **COMNTAUTT, CMTIXAUTT, DROPAUTT**

Privcode 97 (x'61'), 274 (x'112'), 73 (x'49')

Does the user or the role associated with the user own the table?

If so, XAPLUPRM must match the owner name passed from Db2 by the XAPLOWNR parameter when XAPLONRT indicates an authorization ID, or XAPLUCHK must match XAPLOWNR and XAPLUCKT must match XAPLONRT.

If not, the user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.database-name.DBADM	DSNADM
Db2-subsystem.SYSDBADM	DSNADM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

CREATE SYNONYM

XAPLPRIV value: **CRTSYAUTT**

Privcode 102 (x'66')

There are no authorization checks (return code 4).

CREATE VIEW

XAPLPRIV value: **CRTVUAUTT**

Privcode 108 (x'6C')

The user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.SYSCTRL This check is bypassed for user tables.	DSNADM
Db2-subsystem.SYSADM	DSNADM
Db2-subsystem.Db2-database-name-1.DBADM *Db2-subsystem.Db2-database-name-2*.DBADM ⋮*Db2-subsystem.Db2-database-name-n*.DBADM	DSNADM DSNADM DSNADM
Db2-subsystem.SYSDBADM This check is bypassed for user tables.	DSNADM

Note: DBADM authority can be used to allow a user to create views. See “CREATE VIEW privilege” on page 47 for more information.

DELETE

XAPLPRIV value: **DELETAUTT**

Privcode 52 (x'34')

Does the user or the role associated with the user own the table?

If so, XAPLUPRM must match the owner name passed from Db2 by the XAPLOWNR parameter when XAPLONRT indicates an authorization ID, or XAPLUCHK must match XAPLOWNR and XAPLUCKT must match XAPLONRT.

If XAPLACAC is enabled (XAPLFLG2 bit 5 is '1'B) and XAPLUCHK is an authid, suppress the ownership check for XAPLUCHK.

If not, the user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.table-qualifier.table-name.DELETE	MDSNTB or GDSNTB
Db2-subsystem.database-name.DBADM	DSNADM
Db2-subsystem.SQLADM This check is bypassed for user tables and SYSIBM.SYSAUDITPOLICIES.	MDSNSM or GDSNSM
Db2-subsystem.SYSDBADM This check is bypassed for user tables and SYSIBM.SYSAUDITPOLICIES.	DSNADM
Db2-subsystem.DATAACCESS This check is bypassed for SYSIBM.SYSAUDITPOLICIES.	DSNADM
Db2-subsystem.ACCESSCTRL This check is bypassed for user tables and SYSIBM.SYSAUDITPOLICIES.	DSNADM
Db2-subsystem.SYSCTRL This check is bypassed for user tables and SYSIBM.SYSAUDITPOLICIES.	DSNADM
Db2-subsystem.SYSADM This check is bypassed for SYSIBM.SYSAUDITPOLICIES only when Separate Security =YES	DSNADM
Db2-subsystem.SECADM This check is bypassed for user tables.	DSNADM

DROP ALIAS

XAPLPRIV value: **DRPALAUTT**

Privcode 20 (x'14')

Does the user or the role associated with the user own the table?

If so, XAPLUPRM must match the owner name passed from Db2 by the XAPLOWNR parameter when XAPLONRT indicates an authorization ID, or XAPLUCHK must match XAPLOWNR and XAPLUCKT must match XAPLONRT.

If not, the user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.SYSDBADM	DSNADM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

DROP SYNONYM

XAPLPRIV value: **DRPSYAUTT**

Privcode 104 (x'68')

There are no authorization checks (return code 4).

INDEX

XAPLPRIV value: **INDEXAUTT**

Privcode 56 (x'38')

Does the user or the role associated with the user own the table?

If so, XAPLUPRM must match the owner name passed from Db2 by the XAPLOWNR parameter when XAPLONRT indicates an authorization ID, or XAPLUCHK must match XAPLOWNR and XAPLUCKT must match XAPLONRT.

If not, the user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.table-qualifier.table-name.INDEX	MDSNTB or GDSNTB
Db2-subsystem.database-name.DBADM	DSNADM
Db2-subsystem.SYSDBADM	DSNADM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

INSERT

XAPLPRIV value: **INSRTAUTT**

Privcode 51 (x'33')

Does the user or the role associated with the user own the table?

If so, XAPLUPRM must match the owner name passed from Db2 by the XAPLOWNR parameter when XAPLONRT indicates an authorization ID, or XAPLUCHK must match XAPLOWNR and XAPLUCKT must match XAPLONRT.

If XAPLACAC is enabled (XAPLFLG2 bit 5 is '1'B) and XAPLUCHK is an authid, suppress the ownership check for XAPLUCHK.

If not, the user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.table-qualifier.table-name.INSERT	MDSNTB or GDSNTB
Db2-subsystem.database-name.DBADM	DSNADM
Db2-subsystem.SQLADM This check is bypassed for user tables and SYSIBM.SYSAUDITPOLICIES.	MDSNSM or GDSNSM
Db2-subsystem.SYSDBADM This check is bypassed for user tables and SYSIBM.SYSAUDITPOLICIES.	DSNADM
Db2-subsystem.DATAACCESS This check is bypassed for SYSIBM.SYSAUDITPOLICIES.	DSNADM
Db2-subsystem.ACCESSCTRL This check is bypassed for user tables and SYSIBM.SYSAUDITPOLICIES.	DSNADM

One of these resources:	In class:
Db2-subsystem.SYSCTRL This check is bypassed for user tables and SYSIBM.SYSAUDITPOLICIES.	DSNADM
Db2-subsystem.SYSADM This check is bypassed for SYSIBM.SYSAUDITPOLICIES only when Separate Security =YES.	DSNADM
Db2-subsystem.SECADM This check is bypassed for user tables.	DSNADM

LOAD

XAPLPRIV value: **LOADAUTT**

Privcode 75 (x'4B')

Does the user or the role associated with the user own the table?

If so, XAPLUPRM must match the owner name passed from Db2 by the XAPLOWNR parameter when XAPLONRT indicates an authorization ID, or XAPLUCHK must match XAPLOWNR and XAPLUCKT must match XAPLONRT.

If not, the user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.database-name.LOAD	MDSNDB or GDSNDB
Db2-subsystem.database-name.DBCTRL	DSNADM
Db2-subsystem.database-name.DBADM	DSNADM
Db2-subsystem.DATAACCESS	DSNADM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

LOCK TABLE

XAPLPRIV value: **LOCKAUTT**

Privcode 98 (x'62')

Does the user or the role associated with the user own the table?

If so, XAPLUPRM must match the owner name passed from Db2 by the XAPLOWNR parameter when XAPLONRT indicates an authorization ID, or XAPLUCHK must match XAPLOWNR and XAPLUCKT must match XAPLONRT.

If not, the user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.table-qualifier.table-name.SELECT	MDSNTB or GDSNTB
Db2-subsystem.database-name.DBADM	DSNADM
Db2-subsystem.DATAACCESS	DSNADM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

REFERENCES

XAPLPRIV value: **REFERAUTT**

Privcode 54 (x'36')

Does the user or the role associated with the user own the table?

If so, XAPLUPRM must match the owner name passed from Db2 by the XAPLOWNR parameter when XAPLONRT indicates an authorization ID, or XAPLUCHK must match XAPLOWNR and XAPLUCKT must match XAPLONRT.

If not, the user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.table-qualifier.table-name.REFERENCES	MDSNTB or GDSNTB
Db2-subsystem.table-qualifier.table-name.ALTER	MDSNTB or GDSNTB
Db2-subsystem.table-qualifier.table-name.column.REFERENCES	MDSNTB or GDSNTB
Db2-subsystem.database-name.DBADM	DSNADM
Db2-subsystem.SYSDBADM	DSNADM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

REFRESH

XAPLPRIV value: **RFRSHAUTT**

Privcode 275 (x'113')

Does the user or the role associated with the user own the table?

If so, XAPLUPRM must match the owner name passed from Db2 by the XAPLOWNR parameter when XAPLONRT indicates an authorization ID, or XAPLUCHK must match XAPLOWNR and XAPLUCKT must match XAPLONRT.

If not, the user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.database-name.DBCTRL	DSNADM
Db2-subsystem.database-name.DBADM	DSNADM
Db2-subsystem.DATAACCESS	DSNADM
Db2-subsystem.SYSCTRL This check is bypassed for user tables.	DSNADM
Db2-subsystem.SYSADM	DSNADM

RENAME INDEX

XAPLPRIV value: **RNIDXAUTT**

Privcode 283 (x'11B')

Does the user or the role associated with the user own the index?

If so, XAPLUPRM must match the owner name passed from Db2 by the XAPLOWNR parameter when XAPLONRT indicates an authorization ID, or XAPLUCHK must match XAPLOWNR and XAPLUCKT must match XAPLONRT.

If not, the user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.database-name.DBMAINT	DSNADM
Db2-subsystem.database-name.DBCTRL	DSNADM
Db2-subsystem.database-name.DBADM	DSNADM
Db2-subsystem.SYSDBADM	DSNADM
Db2-subsystem.database-name.SYSCTRL	DSNADM
Db2-subsystem.database-name.SYSADM	DSNADM

RENAME TABLE

XAPLPRIV value: **RNTABAUTT**

Privcode 251 (x'FB')

Does the user or the role associated with the user own the table?

If so, XAPLUPRM must match the owner name passed from Db2 by the XAPLOWNR parameter when XAPLONRT indicates an authorization ID, or XAPLUCHK must match XAPLOWNR and XAPLUCKT must match XAPLONRT.

If not, the user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.database-name.DBMAINT	DSNADM
Db2-subsystem.database-name.DBCTRL	DSNADM
Db2-subsystem.database-name.DBADM	DSNADM
Db2-subsystem.SYSDBADM	DSNADM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

SELECT

XAPLPRIV value: **SELCTAUTT**

Privcode 50 (x'32')

Does the user or the role associated with the user own the table?

If so, XAPLUPRM must match the owner name passed from Db2 by the XAPLOWNR parameter when XAPLONRT indicates an authorization ID, or XAPLUCHK must match XAPLOWNR and XAPLUCKT must match XAPLONRT.

If XAPLACAC is enabled (XAPLFLG2 bit 5 is '1'B) and XAPLUCHK is an authid, suppress the ownership check for XAPLUCHK.

If not, the user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.table-qualifier.table-name.SELECT	MDSNTB or GDSNTB
Db2-subsystem.database-name.DBADM	DSNADM

One of these resources:	In class:
Db2-subsystem.SQLADM This check is bypassed for user tables.	MDSNSM or GDSNSM
Db2-subsystem.SYSDBADM This check is bypassed for user tables.	DSNADM
Db2-subsystem.DATAACCESS	DSNADM
Db2-subsystem.ACCESSCTRL This check is bypassed for user tables.	DSNADM
Db2-subsystem.SYSCTRL This check is bypassed for user tables.	DSNADM
Db2-subsystem.SYSADM	DSNADM
Db2-subsystem.SECADM This check is bypassed for user tables.	DSNADM

TRIGGER

XAPLPRIV value: **TRIGAUTT**

Privcode 55 (x'37')

Does the user or the role associated with the user own the table?

If so, XAPLUPRM must match the owner name passed from Db2 by the XAPLOWNR parameter when XAPLONRT indicates an authorization ID, or XAPLUCHK must match XAPLOWNR and XAPLUCKT must match XAPLONRT.

If not, the user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.table-qualifier.table-name.TRIGGER	MDSNTB or GDSNTB
Db2-subsystem.table-qualifier.table-name.ALTER	MDSNTB or GDSNTB
Db2-subsystem.database-name.DBADM	DSNADM
Db2-subsystem.SYSDBADM	DSNADM
Db2-subsystem.SYSCTRL This check is bypassed for user tables.	DSNADM
Db2-subsystem.SYSADM	DSNADM

UPDATE

XAPLPRIV value: **UPDTEAUTT**

Privcode 53 (x'35')

Does the user or the role associated with the user own the table?

If so, XAPLUPRM must match the owner name passed from Db2 by the XAPLOWNR parameter when XAPLONRT indicates an authorization ID, or XAPLUCHK must match XAPLOWNR and XAPLUCKT must match XAPLONRT.

If XAPLACAC is enabled (XAPLFLG2 bit 5 is '1'B) and XAPLUCHK is an authid, suppress the ownership check for XAPLUCHK.

If not, the user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.table-qualifier.table-name.UPDATE	MDSNTB or GDSNTB
Db2-subsystem.table-qualifier.table-name.column.UPDATE	MDSNTB or GDSNTB
Db2-subsystem.database-name.DBADM	DSNADM
Db2-subsystem.SQLADM This check is bypassed for user tables and SYSIBM.SYSAUDITPOLICIES.	MDSNSM or GDSNSM
Db2-subsystem.SYSDBADM This check is bypassed for user tables and SYSIBM.SYSAUDITPOLICIES.	DSNADM
Db2-subsystem.DATAACCESS This check is bypassed for SYSIBM.SYSAUDITPOLICIES.	DSNADM
Db2-subsystem.ACCESSCTRL This check is bypassed for user tables and SYSIBM.SYSAUDITPOLICIES.	DSNADM
Db2-subsystem.SYSCTRL This check is bypassed for user tables and SYSIBM.SYSAUDITPOLICIES.	DSNADM
Db2-subsystem.SYSADM This check is bypassed for SYSIBM.SYSAUDITPOLICIES when Separate Security =YES	DSNADM
Db2-subsystem.SECADM This check is bypassed for user tables.	DSNADM

Any of the table privileges

XAPLPRIV value: **ANYTBAUTT**

Privcode 233 (x'E9')

Does the user or the role associated with the user own the table?

If so, XAPLUPRM must match the owner name passed from Db2 by the XAPLOWNR parameter when XAPLONRT indicates an authorization ID, or XAPLUCHK must match XAPLOWNR and XAPLUCKT must match XAPLONRT.

If not, the user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.table-qualifier.table-name.REFERENCES	MDSNTB or GDSNTB
Db2-subsystem.table-qualifier.table-name.ALTER	MDSNTB or GDSNTB
Db2-subsystem.table-qualifier.table-name.INDEX	MDSNTB or GDSNTB
Db2-subsystem.table-qualifier.table-name.SELECT	MDSNTB or GDSNTB
Db2-subsystem.table-qualifier.table-name.INSERT	MDSNTB or GDSNTB

One of these resources:	In class:
Db2-subsystem.table-qualifier.table-name.DELETE	MDSNTB or GDSNTB
Db2-subsystem.table-qualifier.table-name.UPDATE	MDSNTB or GDSNTB
Db2-subsystem.EXPLAIN	MDSNSM or GDSNSM
Db2-subsystem.database-name.DBADM	DSNADM
Db2-subsystem.SQLADM	MDSNSM or GDSNSM
Db2-subsystem.SYSDBADM	DSNADM
Db2-subsystem.DATAACCESS	DSNADM
Db2-subsystem.ACCESSCTRL This check is bypassed for user tables.	DSNADM
Db2-subsystem.SYSCTRL This check is bypassed for user tables.	DSNADM
Db2-subsystem.SYSADM	DSNADM
Db2-subsystem.SECADM This check is bypassed for user tables.	DSNADM

Table space privileges

Resources: Table spaces

Resource type: R

Note: Having a database privilege on database DSNDB04 is the equivalent of having the privilege on any implicit database.

Db2 privileges

DROP, ALTER

XAPLPRIV values: **DROPAUTR, ALTERAUTR**

Privcode 73 (x'49'), 61 (x'3D')

If the database was created implicitly, and the user or the role associated with the user owns the "other object" (XAPLUPRM is equal to XAPLOON when XAPLOOOT indicates an authorization ID, or XAPLUCHK is equal to XAPLOON and XAPLUCKT is equal to XAPLOOT), access is allowed.

If not, the user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.database-name.DBADM	DSNADM
Db2-subsystem.SYSDBADM	DSNADM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

Note: If the database was created implicitly, *database-name* must be DSNDB04, not the name of the implicit database.

USE

XAPLPRIV value: **USEAUTR**

Privcode 87 (x'57')

The user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.database-name.tablespace-name.USE	MDSNTS or GDSNTS
Db2-subsystem.database-name.DBADM	DSNADM
Db2-subsystem.SYSDBADM	DSNADM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

Trusted context privileges

Resources: Trusted contexts

Resource type: N

Db2 privileges

ALTER TRUSTED CONTEXT

XAPLPRIV value: **ALTERAUTN**

Privcode 61 (x'3D')

The user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.SYSADM Bypass if Separate Security = Yes	DSNADM
Db2-subsystem.SECADM	DSNADM

COMMENT ON TRUSTED CONTEXT

XAPLPRIV value: **COMNTAUTN**

Privcode 97 (x'61')

Does the user or the role associated with the user own the trusted context?

If so, XAPLUPRM must match the owner name passed from Db2 by the XAPLOWNR parameter when XAPLONRT indicates an authorization ID, or XAPLUCHK must match XAPLOWNR and XAPLUCKT must match XAPLONRT.

If not, the user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.SYSCTRL Bypass if Separate Security = Yes	DSNADM

One of these resources:	In class:
Db2-subsystem.SYSADM Bypass if Separate Security = Yes	DSNADM
Db2-subsystem.SECADM	DSNADM

CREATE TRUSTED CONTEXT

XAPLPRIV value: **CREATAUTN**

Privcode 271 (x'10F')

The user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.SYSADM Bypass if Separate Security = Yes	DSNADM
Db2-subsystem.SECADM	DSNADM

DROP TRUSTED CONTEXT

XAPLPRIV value: **DROPAUTN**

Privcode 73 (x'49')

Does the user or the role associated with the user own the trusted context?

If so, XAPLUPRM must match the owner name passed from Db2 by the XAPLOWNR parameter when XAPLONRT indicates an authorization ID, or XAPLUCHK must match XAPLOWNR and XAPLUCKT must match XAPLONRT.

If not, the user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.SYSCTRL Bypass if Separate Security = Yes	DSNADM
Db2-subsystem.SYSADM Bypass if Separate Security = Yes	DSNADM
Db2-subsystem.SECADM	DSNADM

User-defined distinct type privileges

Resources: User-defined distinct types

Resource type: E

Db2 privileges

USAGE

XAPLPRIV value: **USAGEAUTE**

Privcode 263 (x'107')

Does the user or the role associated with the user own the user-defined distinct type?

If so, XAPLUPRM must match the owner name passed from Db2 by the XAPLOWNR parameter when XAPLONRT indicates an authorization ID, or XAPLUCHK must match XAPLOWNR and XAPLUCKT must match XAPLONRT.

If not, the user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.schema-name.type-name.USAGE	MDSNUT or GDSNUT
Db2-subsystem.SYSDBADM	DSNADM
Db2-subsystem.DATAACCESS	DSNADM
Db2-subsystem.SYSADM	DSNADM

User-defined function privileges

Resources: User-defined functions

Resource type: F

Db2 privileges

DISPLAY

XAPLPRIV value: **DISPAUTF**

Privcode 267 (x'10B')

Does the user match the schema name?

If so, XAPLUPRM or XAPLUCHK must match the schema name passed from Db2 by the XAPLOWNQ parameter.

If not, does the user or the role associated with the user own the user-defined function?

If so, XAPLUPRM must match the owner name passed from Db2 by the XAPLOWNR parameter when XAPLONRT indicates an authorization ID, or XAPLUCHK must match XAPLOWNR and XAPLUCKT must match XAPLONRT.

If XAPLACAC is enabled (XAPLFLG2 bit 5 is '1'B) and XAPLUCHK is an authid, suppress the ownership check for XAPLUCHK.

If not, the user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.schema-name.function-name.DISPLAY	MDSNUF or GDSNUF
Db2-subsystem.SYSDBADM	DSNADM
Db2-subsystem.SYSOPR	DSNADM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

EXECUTE

XAPLPRIV value: **CHKEXECF**

Privcode 64 (x'40')

Does the user or the role associated with the user own the user-defined function?

If so, XAPLUPRM must match the owner name passed from Db2 by the XAPLOWNR parameter when XAPLONRT indicates an authorization ID, or XAPLUCHK must match XAPLOWNR and XAPLUCKT must match XAPLONRT.

If not, the user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.schema-name.function-name.EXECUTE	MDSNUF or GDSNUF
Db2-subsystem.SQLADM This check is only done for system defined routines.	MDSNSM or GDSNSM
Db2-subsystem.SYSDBADM This check is only done for system defined routines.	DSNADM
Db2-subsystem.DATAACCESS	DSNADM
Db2-subsystem.SYSADM	DSNADM

START

XAPLPRIV value: **STRTAUTF**

Privcode 265 (x'109')

Does the user match the schema name?

If so, XAPLUPRM or XAPLUCHK must match the schema name passed from Db2 by the XAPLOWNQ parameter.

If not, does the user or the role associated with the user own the user-defined function?

If so, XAPLUPRM must match the owner name passed from Db2 by the XAPLOWNR parameter when XAPLONRT indicates an authorization ID, or XAPLUCHK must match XAPLOWNR and XAPLUCKT must match XAPLONRT.

If not, the user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.SYSDBADM	DSNADM
Db2-subsystem.SYSOPR	DSNADM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

STOP

XAPLPRIV value: **STPAUTF**

Privcode 266 (x'10A')

Does the user match the schema name?

If so, XAPLUPRM or XAPLUCHK must match the schema name passed from Db2 by the XAPLOWNQ parameter.

If not, does the user or the role associated with the user own the user-defined function?

If so, XAPLUPRM must match the owner name passed from Db2 by the XAPLOWNR parameter when XAPLONRT indicates an authorization ID, or XAPLUCHK must match XAPLOWNR and XAPLUCKT must match XAPLONRT.

If not, the user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.SYSDBADM	DSNADM
Db2-subsystem.SYSOPR	DSNADM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

View privileges

Resources: Views

Resource type: V

Db2 privileges

ALTER

XAPLPRIV value: **ALTERAUTV**

Privcode 61 (x'3D')

Does the user or the role associated with the user own the view?

If so, XAPLUPRM must match the owner name passed from Db2 by the XAPLOWNR parameter when XAPLONRT indicates an authorization ID, or XAPLUCHK must match XAPLOWNR and XAPLUCKT must match XAPLONRT.

If not, the user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.SYSDBADM	DSNADM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

COMMENT ON

XAPLPRIV value: **COMNTAUTV**

Privcode 97 (x'61')

Does the user or the role associated with the user own the view?

If so, XAPLUPRM must match the owner name passed from Db2 by the XAPLOWNR parameter when XAPLONRT indicates an authorization ID, or XAPLUCHK must match XAPLOWNR and XAPLUCKT must match XAPLONRT.

If not, the user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.SYSDBADM	DSNADM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

DELETE

XAPLPRIV value: **DELETAUTV**

Privcode 52 (x'34')

Is the view updatable or read-only created from a single table?

If so, does the user or the role associated with the user own the table? This is determined by checking the "other object owner" (XAPLOOON) and "other object owner type" (XAPLOOOT) fields. XAPLOOOT contains an L if the owner is a role and a blank if the owner is not a role. These values must match the corresponding authorization ID values in XAPLUCHK (authorization ID) and XAPLUCKT (type of authorization ID). In addition, If XAPLOOOT is a blank (XAPLOOON is not a role), then if XAPLUPRM matches XAPLOOON, the user owns the table.

If XAPLACAC is enabled (XAPLFLG2 bit 5 is '1'B) and XAPLUCHK is an authid, suppress the ownership check for XAPLUCHK.

If not, the user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.table-qualifier.table-name.view-qualifier.view-name.DELETE	MDSNTB or GDSNTB
Db2-subsystem.database-name.DBADM	DSNADM
Db2-subsystem.DATAACCESS	DSNADM
Db2-subsystem.SYSADM	DSNADM

Note:

1. *table-qualifier*, *table-name*, and *database-name* are for the base table of the view.
2. For an implicit database, *database-name* is DSNDB04.

Is the view created from multiple tables or views)?

If so, the user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.DATAACCESS	DSNADM
Db2-subsystem.view-qualifier.view-name.DELETE	MDSNTB or GDSNTB
Db2-subsystem.SYSADM	DSNADM

DROP

XAPLPRIV value: **DROPAUTV**

Privcode 73 (x'49')

Does the user or the role associated with the user own the view?

If so, XAPLUPRM must match the owner name passed from Db2 by the XAPLOWNR parameter when XAPLONRT indicates an authorization ID, or XAPLUCHK must match XAPLOWNR and XAPLUCKT must match XAPLONRT.

If not, the user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.SYSDBADM	DSNADM
Db2-subsystem.SYSCTRL	DSNADM

One of these resources:	In class:
Db2-subsystem.SYSADM	DSNADM

INSERT

XAPLPRIV value: **INSRTAUTV**

Privcode 51 (x'33')

Is the view updatable (for example, a view created from a single table)?

If so, does the user or the role associated with the user own the table? This is determined by checking the "other object owner" (XAPLOOON) and "other object owner type" (XAPLOOOT) fields. XAPLOOOT contains an L if the owner is a role and a blank if the owner is not a role. These values must match the corresponding authorization ID values in XAPLUCHK (authorization ID) and XAPLUCKT (type of authorization ID). In addition, If XAPLOOOT is a blank (XAPLOOON is not a role), then if XAPLUPRM matches XAPLOOON, the user owns the table.

If XAPLACAC is enabled (XAPLFLG2 bit 5 is '1'B) and XAPLUCHK is an authid, suppress the ownership check for XAPLUCHK.

If not, the user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.table-qualifier.table-name.view-qualifier.view-name.INSERT	MDSNTB or GDSNTB
Db2-subsystem.database-name.DBADM	DSNADM
Db2-subsystem.DATAACCESS	DSNADM
Db2-subsystem.SYSADM	DSNADM

Note:

1. *table-qualifier*, *table-name*, and *database-name* are for the base table of the view.
2. For an implicit database, *database-name* is DSNDB04.

Is the view a read-only view (for example, created from multiple tables)?

If so, the user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.view-qualifier.view-name.INSERT	MDSNTB or GDSNTB
Db2-subsystem.DATAACCESS	DSNADM
Db2-subsystem.SYSADM	DSNADM

INSTEAD OF TRIGGER

XAPLPRIV value: **TRIGAUTV**

Privcode 55 (x'37')

Does the user or the role associated with the user own the view?

If so, XAPLUPRM must match the owner name passed from Db2 by the XAPLOWNR parameter when XAPLONRT indicates an authorization ID, or XAPLUCHK must match XAPLOWNR and XAPLUCKT must match XAPLONRT.

If not, the user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.SYSDBADM	DSNADM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

REGENERATE VIEW

XAPLPRIV value: **ALTERAUTV**

Privcode 61 (x'3D')

Does the user or the role associated with the user own the view?

If so, XAPLUPRM must match the owner name passed from Db2 by the XAPLOWNR parameter when XAPLONRT indicates an authorization ID, or XAPLUCHK must match XAPLOWNR and XAPLUCKT must match XAPLONRT.

If not, the user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.SYSDBADM	DSNADM
Db2-subsystem.SYSCTRL	DSNADM
Db2-subsystem.SYSADM	DSNADM

SELECT

XAPLPRIV value: **SELCTAUTV**

Privcode 50 (x'32')

The user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.view-qualifier.view-name.SELECT	MDSNTB or GDSNTB
Db2-subsystem.DATAACCESS	DSNADM
Db2-subsystem.SYSADM	DSNADM

UPDATE

XAPLPRIV value: **UPDTEAUTV**

Privcode 53 (x'35')

Is the view updatable (for example, a view created from a single table)?

If so, does the user or the role associated with the user own the table? This is determined by checking the "other object owner" (XAPLOOON) and "other object owner type" (XAPLOOOT) fields. XAPLOOOT contains an L if the owner is a role and a blank if the owner is not a role. These values must match the corresponding authorization ID values in XAPLUCHK (authorization ID) and XAPLUCKT (type of authorization ID). In addition, If XAPLOOOT is a blank (XAPLOOON is not a role), then if XAPLUPRM matches XAPLOOON, the user owns the table.

If XAPLACAC is on (XAPLFLG2 bit 5 is '1'B), and XAPLUCHK is an authid, suppress the ownership check for XAPLUCHK.

If not, the user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.table-qualifier.table-name.view-qualifier.view-name.UPDATE	MDSNTB or GDSNTB
Db2-subsystem.table-qualifier.table-name.column-name.view-qualifier.view-name.UPDATE	MDSNTB or GDSNTB
Db2-subsystem.database-name.DBADM	DSNADM
Db2-subsystem.DATAACCESS	DSNADM
Db2-subsystem.SYSADM	DSNADM

Note:

1. *table-qualifier*, *table-name*, *column-name*, and *database-name* are for the base table of the view.
2. For an implicit database, *database-name* is DSNDB04.

Is the view a read-only view (for example, created from multiple tables)?

If so, the user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.view-qualifier.view-name.UPDATE	MDSNTB or GDSNTB
Db2-subsystem.view-qualifier.view-name.column-name.UPDATE	MDSNTB or GDSNTB
Db2-subsystem.DATAACCESS	DSNADM
Db2-subsystem.SYSADM	DSNADM

"Any table" authority

XAPLPRIV value: **ANYTBAUTV**

Privcode 233 (x'E9')

The user must have sufficient authority to:

One of these resources:	In class:
Db2-subsystem.view-qualifier.view-name.SELECT	MDSNTB or GDSNTB
Db2-subsystem.view-qualifier.view-name.INSERT	MDSNTB or GDSNTB
Db2-subsystem.view-qualifier.view-name.UPDATE	MDSNTB or GDSNTB
Db2-subsystem.view-qualifier.view-name.DELETE	MDSNTB or GDSNTB
Db2-subsystem.EXPLAIN	MDSNSM or GDSNSM
Db2-subsystem.SQLADM	MDSNSM or GDSNSM
Db2-subsystem.SYSDBADM	DSNADM
Db2-subsystem.DATAACCESS	DSNADM
Db2-subsystem.ACCESSCTRL This check is bypassed for user tables.	DSNADM
Db2-subsystem.SYSCTRL This check is bypassed when bit 7 of XAPLFLG1 (XAPLUTB) is on.	DSNADM
Db2-subsystem.SYSADM	DSNADM

One of these resources:	In class:
Db2-subsystem.SECADM	DSNADM

This check is bypassed when bit 7 of XAPLFLG1 (XAPLUTB) is on.

Chapter 16. Db2 RACF access control module messages

IRR900A **RACF/DB2 EXTERNAL SECURITY MODULE FAILED TO INITIALIZE FOR DB2 SUBSYSTEM *subsystem-name* BECAUSE CLASS *classname* COULD NOT BE RACLISTED. RACROUTE RETURN CODE *return_code*, RACF RETURN CODE *return_code*, REASON CODE *reason_code*.**

Explanation

The RACF access control module initialization function for Db2 subsystem *subsystem-name* attempted to RACLIST class *classname* using RACROUTE REQUEST=LIST,ENVIR=CREATE,GLOBAL=YES. In a Db2 data sharing environment, *subsystem-name* is the group attachment name. Otherwise, it is the name of the Db2 subsystem. The RACROUTE request failed with the return and reason codes provided in the message text. The return and reason codes are shown in hexadecimal format.

System action

See System Action for message IRR912I or IRR913I.

Operator response

Contact the system programmer.

System programmer response

Use the RACROUTE return code and RACF return and reason codes to determine the cause of the failure. After you correct the problem, restart Db2.

Routing code

Descriptor code is 2. Routing codes are 1 and 9.

IRR901A **RACF/Db2 EXTERNAL SECURITY MODULE FAILED TO INITIALIZE FOR Db2 SUBSYSTEM *subsystem-name* BECAUSE NO ACTIVE Db2 RELATED CLASSES WERE FOUND.**

Explanation

The RACF access control module initialization function for subsystem *subsystem-name* determined that no classes for the indicated Db2 subsystem are active. In a Db2 data sharing environment, *subsystem-name* is the group attachment name. Otherwise, it is the name of the Db2 subsystem.

System action

See System Action for message IRR912I or IRR913I.

Operator response

Contact your security administrator.

Security Administrator Response: Activate the desired classes for the indicated Db2 subsystem and restart Db2.

Routing code

Descriptor code is 2. Routing codes are 1 and 9.

IRR902A **RACF/Db2 EXTERNAL SECURITY MODULE FAILED TO INITIALIZE FOR Db2 SUBSYSTEM *subsystem-name* BECAUSE THE INPUT ACEE WAS {MISSING | NOT VALID}.**

Explanation

The RACF access control module initialization function for subsystem *subsystem-name* determined that the input Db2 subsystem ACEE was either not valid or missing.In a Db2 data sharing environment, *subsystem-name* is the group attachment name. Otherwise, it is the name of Db2 subsystem.

System action

See System Action for message IRR912I or IRR913I.

Operator response

Contact the Db2 system programmer.

System programmer response

Contact IBM Support.

Routing code

Descriptor code is 2. Routing codes are 1 and 9.

IRR903A **RACF/DB2 EXTERNAL SECURITY MODULE FAILED TO INITIALIZE**

FOR DB2 SUBSYSTEM *subsystem-name* BECAUSE RACF WAS NOT ACTIVE.

Explanation

The RACF access control module initialization function for subsystem *subsystem-name* determined that RACF is not active on this system.In a Db2 data sharing environment, *subsystem-name* is the group attachment name. Otherwise, it is the name of the Db2 subsystem.

System action

See System Action for message IRR912I or IRR913I.

Operator response

Contact the RACF system programmer.

System programmer response

Determine why RACF is inactive. After you correct the problem, activate RACF and restart Db2.

Problem determination

Issue the RVARY LIST command to determine RACF status.

Routing code

Descriptor code is 2. Routing codes are 1 and 9.

IRR904I **RACF/Db2 EXTERNAL SECURITY MODULE INITIALIZED WITH WARNINGS FOR Db2 SUBSYSTEM *subsystem-name* BECAUSE A DEFAULT ACEE COULD NOT BE CREATED. RACROUTE RETURN CODE *return_code*, RACF RETURN CODE *return_code*, REASON CODE *reason_code*.**

Explanation

The RACF access control module initialization function for subsystem *subsystem-name* attempted to create a default ACEE to use in subsequent authority checking when no ACEE is provided. In a Db2 data sharing environment, *subsystem-name* is the group attachment name. Otherwise, it is the name of the Db2 subsystem.

The attempt to create the ACEE using RACROUTE REQUEST=VERIFY,ENVIR=CREATE failed with the return and reason codes provided in the message text. The return and reason codes are shown in hexadecimal format.

System action

Processing continues and the RACF access control module is used for subsequent authority checking if Db2 provides an ACEE. If no ACEE is provided, requests are deferred to Db2.

Operator response

Contact the Db2 system programmer.

System programmer response

Use the RACROUTE return code and RACF return and reason codes to determine the cause of the failure. After you correct the problem, restart Db2.

Routing code

Descriptor code is 12. Routing codes are 2, 9, and 10.

IRR905I **RACF/Db2 TERMINATION FUNCTION COMPLETED WITH WARNINGS FOR Db2 SUBSYSTEM *subsystem-name* BECAUSE CLASS *classname* COULD NOT BE UN-RACLISTED. RACROUTE RETURN CODE *return_code*, RACF RETURN CODE *return_code*, REASON CODE *reason_code*.**

Explanation

The RACF access control module termination function for subsystem *subsystem-name* attempted to delete RACLISTed profiles for class *classname*. In a Db2 data sharing environment, *subsystem-name* is the group attachment name. Otherwise, it is the name of the Db2 subsystem.

The attempt to delete the profiles using RACROUTE REQUEST=LIST,ENVIR=DELETE failed with the return and reason codes provided in the message text. The return and reason codes are in hexadecimal format.

System action

The termination function continues processing. Resources are cleaned up when processing completes. This does not impact RACF authorization checking when Db2 is restarted.

Operator response

Contact the Db2 system programmer.

System programmer response

Use the RACROUTE return code and the RACF return and reason codes to determine the cause of the failure.

Routing code

Descriptor code is 12. Routing codes are 2, 9, and 10.

IRR906I **RACF/Db2 TERMINATION FUNCTION COMPLETED WITH WARNINGS FOR Db2 SUBSYSTEM *subsystem-name* BECAUSE THE DEFAULT ACEE COULD NOT BE DELETED. RACROUTE RETURN CODE *return_code*, RACF RETURN CODE *return_code*, REASON CODE *reason_code*.**

Explanation

The RACF access control module termination function for the subsystem *subsystem-name* attempted to delete the default ACEE used by the RACF access control module. In a Db2 data sharing environment, *subsystem-name* is the group attachment name. Otherwise, it is the name of the Db2 subsystem.

The attempt to delete the ACEE using RACROUTE REQUEST=VERIFY,ENVIR=DELETE failed with the return and reason codes provided in the message text. The return and reason codes are in hexadecimal format.

System action

The termination function continues processing and resources are cleaned up when processing completes. This does not impact RACF authorization checking when Db2 is restarted.

Operator response

Contact the Db2 system programmer.

System programmer response

Use the RACROUTE return code and the RACF return and reason codes to determine the cause of the failure. After you correct the problem, restart Db2.

Routing code

Descriptor code is 12. Routing codes are 2, 9, and 10.

IRR907I **RACF/DB2 TERMINATION FUNCTION COMPLETED WITH WARNINGS FOR DB2 SUBSYSTEM *subsystem-name* BECAUSE THE INPUT ACEE WAS {MISSING | NOT VALID}.**

Explanation

The RACF access control module termination function for the subsystem *subsystem-name* determined that the input Db2 subsystem ACEE was either not valid or missing. In a Db2 data sharing environment, *subsystem-name* is the group attachment name. Otherwise, it is the name of the Db2 subsystem.

System action

For exit termination, the RACF access control module is not able to complete its termination function. This should not impact RACF authorization checking when Db2 is restarted.

Operator response

Contact the Db2 system programmer.

System programmer response

Contact IBM Support.

Routing code

Descriptor code is 12. Routing codes are 2, 9, and 10.

IRR908I **RACF/DB2 EXTERNAL SECURITY MODULE FOR DB2 SUBSYSTEM *subsystem-name* HAS A MODULE VERSION OF *module-version* AND A MODULE LENGTH OF *module-length*.**

Explanation

The RACF access control module initialization function for subsystem *subsystem-name* has determined the version and length of the RACF access control module for subsystem *subsystem-name*. In a Db2 data sharing environment, *subsystem-name* is the group attachment name. Otherwise, it is the name of the Db2 subsystem. *module-version* is the FMID or APAR number associated with the module. *module-length* is the hexadecimal length of all CSECTs contained in the module.

System action

The RACF access control module continues.

Routing code

Descriptor code is 4. Routing codes are 9 and 10.

IRR909I **RACF/DB2 EXTERNAL SECURITY MODULE FOR DB2 SUBSYSTEM *subsystem-name* IS USING OPTIONS: &CLASSOPT= *classopt* &CLASSNMT= *classnmt* &CHAROPT= *charopt* &ERROROPT= *erroropt* &PCELLCT= *pcellct* &SCELLCT= *scellct***

Explanation

The RACF access control module initialization function for subsystem *subsystem-name* lists the options that are being used for the RACF access control module. In a Db2 data sharing environment, *subsystem-name* is the group attachment name. Otherwise, it is the name of the Db2 subsystem.

System action

The RACF access control module continues.

Routing code

Descriptor code is 4. Routing codes are 9 and 10.

IRR910I **RACF/DB2 EXTERNAL SECURITY MODULE FOR DB2 SUBSYSTEM *subsystem-name* INITIATED RACLIST FOR CLASSES: {*classname-list* | * NONE *}**

Explanation

The RACF access control module initialization function for Db2 subsystem *subsystem-name* issued a RACROUTE REQUEST=LIST,GLOBAL=YES macro for classes *classname-list* as defined in the object table in the RACF access control module. If * NONE * is displayed, an error occurred before the initialization function could issue RACROUTE REQUEST=LIST for any class. In a Db2 data sharing environment, *subsystem-name* is the group attachment name. Otherwise, it is the name of the Db2 subsystem.

System action

The RACF access control module continues.

Routing code

Descriptor code is 4. Routing codes are 9 and 10.

IRR911I **RACF/DB2 EXTERNAL SECURITY MODULE FOR DB2 SUBSYSTEM *subsystem-name* SUCCESSFULLY RACLISTED CLASSES: {*classname-list* | * NONE *}**

Explanation

The RACF access control module initialization function for Db2 subsystem *subsystem-name* lists the classes for which the RACROUTE REQUEST=LIST,GLOBAL=YES macro was successful.If * NONE * is displayed, no classes were RACLISTed successfully. See message IRR910I to determine which classes the RACF access control module attempted to use. The class list displayed in IRR911I might be a valid subset of the classes listed in message IRR910I.

System action

The RACF access control module continues.

Routing code

Descriptor code is 4. Routing codes are 9 and 10.

IRR912I **NATIVE DB2 AUTHORIZATION IS USED.**

Explanation

RACF is not being used to control access to Db2 resources. This message is preceded by other messages that describe why RACF is not being used for access control decisions.

System action

None. All subsequent access control decisions are made by the Db2 using Db2 native security mechanism.

Operator response

Follow the Operator Response for the message that preceded this message.

Routing code

Descriptor code is 2. Routing codes are 1 and 9.

IRR913I **DB2 SUBSYSTEM TERMINATION REQUESTED.**

Explanation

RACF has requested that the Db2 subsystem be terminated. This message is preceded by another message that describes why this request has been made.

System action

RACF has requested that the Db2 subsystem terminate.

Operator response

Follow the Operator Response for the message that preceded this message.

Routing code

Descriptor code is 2. Routing codes are 1 and 9.

IRR914I **The RACF/DB2 external security module has been invoked with a DB2 VxRxMx parameter list**

Explanation

The RACF access control module was invoked, but the parameter list that was passed was for a different version of Db2. This mismatch of Db2 version and level of the RACF access control module is not allowed.

System action

If the RACF access control module has installation option &ERR0R0PT 2 specified, then the Db2 subsystem is asked to terminate. If installation option &ERR0R0PT 1 was specified, then the Db2 subsystem is asked to use native Db2 authorization. In either case, the exit is not called again.

System programmer response

Db2 must run with the RACF/Db2 external security module that was shipped with Db2. The Db2 version must be assembled with the Db2 macros, link-edited, and installed in a library that is accessible to your Db2 subsystem.

Routing code

Descriptor code is 12. Routing codes are 2, 9, and 10.

IRR915I **EXPLRC1 = *xxx*, EXPLRC2 = *xxx*, XAPLPRIV = *xxxx***

Explanation

The RACF access control module has been instructed (either by a zap or by changing the assembler source) to display the return and reason code (EXPLRC1 and EXPLRC2) that is returned to Db2 along with the Db2 privilege code (XAPLPRIV) for the request. For Db2 initialization and termination, XAPLPRIV is *xxx*.

System action

None. This message is a diagnostic informational message.

System programmer response

None. This message is only issued if the RACF access control module has been specifically altered to display the return, reason, and privilege codes. This alteration should only be done under the guidance of the IBM service team.

Routing code

Descriptor code is 4. Routing codes are 9 and 10.

IRR916I **RACF/Db2 EXTERNAL SECURITY MODULE WAS ASSEMBLED WITH AN [HRF7720 OR EARLIER | HRF7730 OR LATER] MACRO LIBRARY. Db2 ROLES AS RACF CRITERIA ARE [NOT] SUPPORTED.**

Explanation

This message is issued when the Db2 V9 RACF access control module is used, to indicate whether or not the module supports Db2 roles.

The module does not fully support Db2 roles if it is invoked from a Db2 V9 system and any of the following sets of conditions are true:

- The system is running z/OS V1R7 and the RACF access control module was assembled with z/OS V1R7 macros.
- The system is running z/OS V1R7 and the RACF access control module was assembled with z/OS V1R8 macros.
- The system is running z/OS V1R8 and the RACF access control module was assembled with z/OS V1R7 macros.

The module fully supports Db2 roles if is invoked from a Db2 V9 system and the following set of conditions is true:

- The system is running z/OS V1R8 (or higher) and the RACF access control module was assembled with z/OS V1R8 (or higher) macros.

System action

The RACF access control module continues.

System programmer response

If the message indicates that Db2 roles as RACF criteria are not supported, and you need this support,

reassemble the RACF access control module with the HRF7730 or later macro library to fully enable support for roles in the module when Db2 is running on z/OS V1R8 or later. The version of the module shipped with Db2 V9 must be assembled with the Db2 V9 macros, link-edited, and installed in a library that is accessible to your Db2 subsystem.

If the message indicates that Db2 roles are supported as RACF criteria, no further action is required.

Routing code

Descriptor code is 4. Routing codes are 9 and 10.

Information resources for Db2 11 for z/OS and related products

Information about Db2 11 for z/OS and products that you might use in conjunction with Db2 11 is available online in IBM Documentation.

You can find the complete set of product documentation for Db2 11 for z/OS in IBM Documentation.

You can also download other PDF format manuals for Db2 11 for z/OS from IBM Documentation in PDF format manuals for Db2 11 for z/OS (Db2 for z/OS in IBM Documentation).

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

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, MD-NC119
Armonk, NY 10504-1785 US

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing Legal and Intellectual Property Law IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

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 jurisdictions 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.

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

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Director of Licensing IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785 US

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

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 actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate 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. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work must include a copyright notice as shown below:

© (*your company name*) (*year*).
Portions of this code are derived from IBM Corp. Sample Programs.
© Copyright IBM Corp. (*enter the year or years*).

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Programming interface information

This information is intended to help you invoke the RACF access control module on Db2 11 for z/OS servers. This information primarily documents Product-sensitive Programming Interface and Associated Guidance Information provided by Db2 11 for z/OS.

Product-sensitive Programming Interface and Associated Guidance Information

Product-sensitive Programming Interfaces allow the customer installation to perform tasks such as diagnosing, modifying, monitoring, repairing, tailoring, or tuning of this IBM software product. Use of such interfaces creates dependencies on the detailed design or implementation of the IBM software product. Product-sensitive Programming Interfaces should be used only for these specialized purposes. Because of their dependencies on detailed design and implementation, it is to be expected that programs written to such interfaces may need to be changed in order to run with new product releases or versions, or as a result of service.

Trademarks

IBM, the IBM logo, and ibm.com® are trademarks or registered marks 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 IBM trademarks is available on the web at "Copyright and trademark information" at: http://www.ibm.com/legal/copytrade.shtml.

Linux® is a registered trademark of Linus Torvalds 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.

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

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions:

Applicability: These terms and conditions are in addition to any terms of use for the IBM website.

Personal use: You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use: You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights: Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

Privacy policy considerations

IBM Software products, including software as a service solutions, (“Software Offerings”) may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user, or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering’s use of cookies is set forth below.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM’s Privacy Policy at http://www.ibm.com/privacy and IBM’s Online Privacy Statement at http://www.ibm.com/privacy/details the section entitled “Cookies, Web Beacons and Other Technologies” and the “IBM Software Products and Software-as-a-Service Privacy Statement” at http://www.ibm.com/software/info/product-privacy.

Glossary

The glossary is available in IBM Knowledge Center.

See the Glossary topic for definitions of Db2 for z/OS terms.

Index

Special Characters

A

B

C

D

E

G

I

L

M

O

P

R

S

T

U

V

W

X

Product Number: 5615-DB2
5697-P43

SC19-4065-06