CA ACF2™ Option for DB2

Product Guide

r1.3

Second Edition

CA Technologies Product References

This document references the following CA Technologies products:

- CA ACF2 for z/OS

Contact CA Technologies

Contact CA Support

For your convenience, CA Technologies provides one site where you can access the information that you need for your Home Office, Small Business, and Enterprise CA Technologies products. At http://ca.com/support, you can access the following resources:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

Providing Feedback About Product Documentation

If you have comments or questions about CA Technologies product documentation, you can send a message to techpubs@ca.com.

To provide feedback about CA Technologies product documentation, complete our short customer survey which is available on the CA Support website at http://ca.com/docs.

Documentation Changes

The following documentation updates have been made since the last release of this documentation:

- Which Resources Require Rule Sets— added content around system DBADM,
- Setting the Level of Protection—added SYSDBADM.
- DB2 Prevalidation Exit (DB2PRE)—added SYSDBADM.
- DB@ Postvalidation Exit (DB2POST)—added SYSDBADM.

Contents

Chapter 1: Introduction 7

Chapter 2: Implementation 9

Chapter 3: Customization 47

Chapter 4: Troubleshooting 53

Chapter 5: Using Secondary Authorization IDs 59

Chapter 6: Using SAF 65

Chapter 7: Checklists 67

Chapter 1: Introduction

This guide introduces you to CA ACF2™ Option for DB2 (CA ACF2 for DB2). By the time you have finished reading this guide, you will have an overview of the wide scope of the product and its usability will be familiar to you. It is important to us that you feel comfortable with CA ACF2 for DB2 before you begin to use it.

This guide describes the process that the implementation team should follow to install and implement CA ACF2 for DB2 and CAIENF. This guide explains how the conversion utility converts SQL GRANT statements to rule entries. It also describes performance and tuning considerations, exit points, and other product implementation information.

Chapter 2: Implementation

This section contains the following topics:

Organize and Plan Your Implementation

One of the most important phases in properly implementing CA ACF2 for DB2 is to organize and plan each step. The implementation plan is the framework for installing and implementing CA ACF2 for DB2. It is key to the success of a properly designed and implemented security system.

To assist you in this process, we suggest you perform the following:

- Create a planning checklist
- Decide on centralized or decentralized security
- Appoint your implementation team
- Plan and coordinate your implementation schedule
- Distribute CA ACF2 documentation

Create Planning Checklist

Use the planning checklist to ensure the smooth flow of installation and implementation events. You can also use it to ensure that nothing is overlooked when you install and implement CA ACF2 for DB2. A sample checklist is provided in the “Checklists” appendix.

Decide on Centralized or Decentralized Security

With native DB2 security, your site may not have a choice as to how you administer security. Because of the way DB2 is designed, your database administrators (DBAs) may be responsible for many of the functions normally performed by security administrators. With CA ACF2 for DB2, you can decide whether you want the security functions and responsibilities for DB2 to be centralized or decentralized. Whether CA ACF2 for DB2 administration is centralized or decentralized (and to what degree) depends on the size, structure, and unique needs of your site. You may want some DBAs to perform a subset of administrative functions or you may want your security department to handle all security concerns.

Central security administrators can administer any CA ACF2 for DB2 rule or record, while security administrators in a decentralized environment have jurisdiction over limited groups of CA ACF2 for DB2 rules, records, and DB2 resources. Security administrators in a centralized environment have full access to DB2 resources, infostorage records (where CA ACF2 for DB2 stores rules and records), and any other resource available to a user with the SECURITY attribute. One of their functions can be to create other security administrators.

To decentralize security, you can limit the authority that security administrators have when they write CA ACF2 for DB2 rules or when they create DB2 records. You can use the scoping feature to impose these restrictions. This feature restricts the authority of security administrators to maintain DB2 records and CA ACF2 for DB2 rules for separate areas of responsibility. To limit a security administrator to CA ACF2 for DB2 information, place a scope record that includes DB2 infostorage records in his logonid record.

The following example shows how to create a security administrator by assigning the SECURITY privilege to his logonid record. It also shows how to scope this security administrator to DB2 infostorage records. (This example assumes a USER01 logonid has already been created.)

```
READY
acf
  ACF
set lid
  LID
change user01 security
```

Then, add an entry in the INF field of a scope record to limit the user to DB2 records and CA ACF2 for DB2 rules. The ACF subcommands to perform this are shown in the following example:

```
set scope(scp)
 SCOPE
change db2scop inf(d-,cdb2-)
```

This example assumes a scope record of DB2SCOP exists.

Associate this scope record with the security administrator’s logonid record, as follows:

```
set lid
 LID
change user01 scplist(db2scop)
```

See the *Administrator Guide* for more details about scoping administrators.

Appoint Your Implementation Team

The main function of the implementation team is to properly implement CA ACF2 and related security systems and procedures. Their function is limited because most of the work occurs during the planning and implementation phases.

If you created an implementation team (IT) to implement CA ACF2, you might want to use this team to implement CA ACF2 for DB2. In addition to the current members, you will want to ask DB2 database administrators to join this team if they were not already included. Their participation and acceptance of CA ACF2 for DB2 is essential to successfully implementing CA ACF2 for DB2, because they play a critical role in creating and maintaining DB2 resources.

Plan and Coordinate Your Implementation Schedule

As the implementation plan takes shape, you should schedule all activities and events, and distribute schedules throughout the site to all key organizations. You should coordinate implementation timetables with related implementations of other system or product installations so they do not conflict. Follow these steps to prepare for implementation.

1. Ensure the key departments and personnel take an active role in the implementation plans. Everyone must support the implementation.
2. Meet with other department managers to determine their concerns. Introduce an approach to address their concerns without compromising security.
3. Observe other knowledgeable workers as they perform their functions. This can help lead to other improvements in securing resources that they use daily.
4. Communicate the scope, impact, and requirements of the implementation plan to the entire organization.
5. Design the CA ACF2 for DB2 installation and security implementation. Be sure to take into consideration the current data center standards, conventions, and procedures for testing and production turnover.
6. Ensure that key personnel understand their roles and are available to support the implementation project.
7. Plan for personnel vacations, holidays, and other staff outages.
8. Secure management approval for the complete implementation plan, objectives, approach, and timetables.

Distribute Documentation

Determine which members of your implementation team (IT) require CA ACF2 for DB2 guides. Ensure that they also have the applicable CA ACF2 guides. You might want to distribute documentation to various groups as shown in the following table:

Group	Guides
All IT members	*Installation Guide* *Administrator Guide* CA ACF2 *for z/OS Implementation Planning Guide* CA ACF2 *for z/OS Administrator Guide* CA ACF2 *for z/OS Reports and Utilities Guide* CA ACF2 *for z/OS General Information Guide*
System Programming IT member	*Installation Guide* *Messages Guide* CA ACF2 *Security for z/OS Systems Programmer Guide* CA ACF2 *for z/OS Installation Guide*

You might want to distribute additional documentation during the latter phases of implementation. For example, you should provide your operations staff with a copy of the *Messages Guide* before the first test.

Provide Adequate Training

Although not necessary, we recommend that you send your key implementation team members to DB2 classes and to CA ACF2 training classes. The client education division of CA offers courses in various cities on a regular basis. They can also provide on-site training for your organization. For more information about available training classes or to request an Education Catalog, call (800) 237-9273.

Review Security Policy

Members of the IT should know the company data security goals and objectives before implementing CA ACF2 for DB2. Your IT members must evaluate the existing security policy and determine whether any modifications are needed. Identify any areas of the policy that are unclear or inappropriate. Keep in mind the modifications that your team wants to make as you plan the implementation in your data center.

Identify Current Security Standards

The first step in reviewing your security policy is to identify existing site security policies or standards. Because not every site has a formal security policy, start by identifying what has already been protected. Then, based on the implementation plan, identify the changes you must make in your current security policy. Note the differences in policy or functions of the features. During the implementation of CA ACF2 for DB2, you can change anything that you did not implement in CA ACF2 the way you originally wanted. Communicate any necessary changes to the user community by publishing a new security policy.

Many factors influence a site's policies and objectives. These can include the following areas:

- Government regulations that can affect data security requirements at your site, such as the National Computer Security Center (NCSC) ratings, the Privacy Act, Foreign Corrupt Practices Act, Securities and Exchange Commission (SEC) and other agency regulations, and various other accounting and reporting requirements.
- Legal requirements, such as controls over Electronic Funds Transfers (EFT) and contractual agreements preventing the disclosure of information.
- Industry practices and agreements, as they relate to recognized standards of due care.
- The threats of sabotage, white-collar crime, and computer fraud.
- Good business practices, such as separation of function, clear lines of responsibility and authority, individual accountability, knowing what the control procedures are and that they are in place, knowing who has access to assets and records and controlling this access, and various auditing considerations.
- Existing corporate policies that relate to computer assets such as data security, access control, and computer control auditability.
- Impact on the user community, such as decisions that relate to the degree of functional separation and the degree of centralization or decentralization in the administration of CA ACF2 for DB2 and related controls.
- Procedure enforcement needs, such as naming conventions for user identification.

Tailor Your New Security Policy

After you evaluate existing security policies and standards, you must tailor your new security policy to accommodate DB2 requirements. If these policies and goals are not defined, the IT will find it difficult to choose appropriate options for CA ACF2 for DB2 and to proceed quickly through CA ACF2 for DB2 implementation. Communicate the new policy to your user community to ensure that they know the new goals. You can use CA ACF2 for DB2 as a tool to implement security policies, automate policy enforcement, and help the company achieve its goals in relation to DB2 security.

As you begin tailoring the security policy, you must make decisions based on the following questions:

- What is the condition of my current security system?
- How do I want to secure development versus production systems?
- Do I want to centralize or decentralize my security administration?
- What role should DBAs play?
- How do I migrate an application from test to production? What issues should I consider?
- How much authority do I want to give the developers of applications?
- What resources do I want to audit?
- Which audit tools do I want to use (TRACE on the logonid, LOG on the rule entry, reports, and so on)?

Identify Your Operating Environment

To select the most appropriate options and to effectively use CA ACF2 for DB2 controls, first identify local conditions that you must consider, such as:

- Local naming conventions
- Existing security mechanisms
- Uniqueness of user identification
- Operating system configuration

Establish Local Naming Conventions

Determine existing (or desired) naming conventions for:

- User identification (logonids and UIDs) for TSO, batch, IMS, CICS, Call Attachment Facility (CAF), and Distributed Data Facility (DDF) users who access DB2
- Names of DB2 subsystems
- Names of DB2 resources (such as databases, tables, table spaces, and so on)

Standard naming conventions for UIDs, DB2 subsystems, and DB2 resources simplify the task of writing CA ACF2 for DB2 rules. These conventions let you use masking characters to identify multiple characters. Instead of creating one rule set for each resource, you can write a single rule that applies to multiple resources. Naming conventions also enable you to write rules before resources are created. Therefore, when you create a table, a rule set that controls access to the table already exists.

For example, to create a table in DB2 and be able to access and update the table, you might require the following privileges:

- CREATETAB on the database
- CREATETS on the database
- USE OF TABLE SPACE
- UPDATE on the table
- SELECT on the table

If you use naming conventions for your tables and table spaces, rule sets with masking characters might already cover these resources. For example, the ability to perform a SELECT on a newly created table named EMPLOYEE.BENEFITS is covered by the following existing rule set:

```
$KEY(EMPLOYEE.******************)
$TYPE(TBL)
$SYSID(****)
 UID(*****PER) SERVICE(SELECT) ALLOW
```

If you use naming conventions, other privileges could be similarly covered by existing rule sets.

Identify Existing Security Mechanisms

Identify all existing security mechanisms such as GRANT and REVOKE statements or security checks that are built into applications and decide which ones to replace with CA ACF2 for DB2. Before implementation you might want to keep current DB2 security mechanisms active. Through the DB2 OPTS infostorage record, you can activate or deactivate CA ACF2 for DB2 security on a subsystem-by-subsystem basis.

Evaluate your site’s use of the System Authorization Facility (SAF). While you can use SAF to validate your connection to the DB2 subsystem, SAF does not validate access to any of the resources in DB2. CA ACF2 for DB2 provides you with a greater degree of control over your DB2 environment. It validates each user’s access to each individual resource. You can instruct CA ACF2 for DB2 to use the following criteria to make access decisions:

- Name of the resource
- Type of resource
- Particular DB2 subsystem
- Environment of the request.

These criteria may make the use of SAF to validate access to the subsystem unnecessary.

Your site, however, might require the use of SAF. If so, the CA ACF2 SAF interface lets you use SAF with CA ACF2. See the “Using SAF” appendix in this guide for more information about using SAF with DB2.

Ensure Uniqueness of User Identifications

Identify all users and any individual or group IDs. Ensure that each system user is positively identified with a unique CA ACF2 logonid and a single password. Ensure that you create appropriate logonids for each secondary authorization ID.

Ensure that you define logonids for all processes that connect to DB2 through another subsystem such as CICS. Through the *resource control table* (RCT), CICS can send DB2 user IDs, operator IDs, terminal names, transaction IDs, group IDs, and user-defined IDs. You must create a logonid for each ID specified in the RCT. See the “Implementation” chapter for more information about creating logonids.

You also might want to reevaluate your user identification string (UID) construction if you want to be able to dynamically alter it. See the Select Options section in this chapter and the CA ACF2 for z/OS *Implementation Planning Guide* for more information about UID construction.

Review Operating System Configuration

Ensure that your site has met the following requirements (see the “System Requirements” section in the “Installation” chapter for specific information about each of these requirements):

- DB2 requirements
- CA ACF2 for DB2 requirements
- CAIENF requirements.

Select Options

This step helps you select the CA ACF2 options that you can tailor to meet your needs. You can select temporary values for some options to phase CA ACF2 for DB2 protection into each of your DB2 subsystems for transition purposes. For example, you can choose to control test subsystems only for the first week, expand to production subsystems during week two, and so on. Other options are more permanent when they are implemented. They require careful thought and planning to ensure that they meet your needs adequately. These options include the use of secondary authorization IDs, construction of your UID string, and the use of exits.

You can indicate temporary values to CA ACF2 for DB2 by setting up various parameters in CA ACF2 and CA ACF2 for DB2. If you have installed CA ACF2 and you want to change certain options to accommodate CA ACF2 for DB2, you can change these options using ACF commands and subcommands. To change or insert the values of DB2 records, you must first install CA ACF2 for DB2 to use ACF commands and subcommands with these records. After you change the values, you must stop and start the DB2 subsystem to make the values effective.

Determine CA ACF2 Control Options

When you installed CA ACF2, you set many of the options that affect CA ACF2 for DB2 processing. Review these options to ensure that you understand their affect on CA ACF2 for DB2 processing. These options include the following GSO records.

INFODIR

This record controls which rule directories and rule sets in the Infostorage database are made resident in storage. This lets you access rule directories and rule sets from common storage instead of DASD, which helps increase the performance of accessing highly used rule sets. You can also use the GSO INFODIR record to make other infostorage records resident. See the *Administrator Guide* for more information about using the GSO INFODIR.

RULEOPTS

The CHANGE field of this record controls whether the %CHANGE and %RCHANGE control statements are recognized in a rule set. The $NOSORT field of this record controls whether the $NOSORT control statement in the rule set is processed.

OPTS

The DATE field of this record defines the date format that is accepted in the UNTIL rule parameter and the format that CA ACF2 for DB2 is used in reporting. The INFOLIST field of this record specifies the logonid attributes a user must have to view records stored in the Infostorage database.

Determine CA ACF2 for DB2 Control Options

DB2 records stored in the Infostorage database control CA ACF2 for DB2 parameters. These records determine the exits that your DB2 subsystem will take and the level of protection given to each of your DB2 subsystems.

EXITS Record

Specify the exits that should gain control of CA ACF2 for DB2 processing before and after CA ACF2 for DB2 performs a validation. Through your exit programs, you can force a request or change the type and name of the DB2 resource used in the validation.

OPTS Record

Specify whether CA ACF2 for DB2 should control a particular DB2 subsystem. If you want CA ACF2 for DB2 to control requests in the subsystem, you can specify the level of protection for each resource in that subsystem, as follows:

Protection Level	Description
ABORT	Prevent access to the resource when a rule entry denies access.
LOG	Allow but log access to the resource even when a rule entry denies access.
QUIET	Allow access to the resource even when a rule entry denies access.
RULE	Follow the access specified in the rule set.

Because the default for processing is ABORT mode, you might want to change the mode in this record to QUIET or LOG when you first start CA ACF2 for DB2. This lets production jobs continue processing, while you gather data about who is accessing what resources, and so on. In ABORT mode, the next applicable rule set is searched to determine access. If no applicable rule set exists or grants access, ABORT causes CA ACF2 for DB2 to suspend jobs that violate the rule set and to log the violation.

If you want to write CA ACF2 for DB2 resource rules that control more than one DB2 subsystem but you cannot appropriately mask the DB2 subsystem IDs in the $SYSID field in the resource rule, use the GSYSID field in the OPTS record. If the GSYSID value is specified in the OPTS record, CA ACF2 for DB2 uses it as the SYSID during resource authorizations rather than the DB2 subsystem ID. You should write your DB2 resource rules using that value as the $SYSID field value.

The “Implementation” chapter explains how to use the DB2 OPTS record to migrate to full security.

See the *Administrator Guide* for information about creating, changing, and deleting DB2 records.

Evaluate Use of Secondary Authorization IDs

Secondary authorization IDs are alternate group IDs with which DB2 users can be associated during DB2 processing. A DB2 user can be assigned one or more secondary authorization IDs in a DB2 connection exit when the user connects to DB2. Some of the reasons IBM created secondary IDS are to overcome the cascade effect and improve the ownership of DB2 objects. The *cascade effect* occurs when DB2 revokes a privilege from an ID that granted the privilege to other IDS. (The granting ID must have the WITH GRANT OPTION to grant the privilege to other IDS.) DB2 then revokes the privilege from the other IDs. The loss of privileges creates an administrative burden for the security administrator, because he must ensure that users have the appropriate privileges to do their jobs. Secondary IDs can create a cushion against this type of effect because they typically are not granted the WITH GRANT OPTION and cannot grant privileges to another ID.

Secondary IDs also address the problem of managing ownership of DB2 objects. When a user is terminated or transferred, the security administrator must revoke his privileges and grant them to another user. This can create administrative overhead. With secondary authorization IDs, the security administrator can grant these privileges to a secondary ID instead. The user then can associate with the secondary ID and exercise the privileges it has. Through the exits, the security administrators must update only a list of who can associate with each secondary ID when a user is transferred or terminated.

Some sites, however, use secondary IDs for other purposes than resource ownership. For example, users running applications that refer to an unqualified table name can use different versions of the table by changing their current SQL ID to a different secondary ID. If your site uses secondary authorization IDs in this way, you can continue to use them with CA ACF2 for DB2 and create reports against each of the IDs (primary, secondary, and original authorization IDs) associated with a user. However, CA ACF2 for DB2 validates secondary authorization IDs in the same way that it validates logonids. This means that you must create a logonid record for each secondary authorization ID and store it in the CA ACF2Logonid database for it to be effective. In addition, you must use CA ACF2-supplied exits and create source group entries to define the secondary ID list. See the “Using Secondary IDs” appendix for more information about these exits and source group entries.

If you use secondary IDs for the above purposes, recognize that they can be difficult to maintain, can increase the complexity of managing DB2 security, and can create significant performance overhead. When DB2 checks a user’s authorization, it checks every ID associated with that user until it finds one that is authorized. If users have multiple secondary IDs, this process can be time-consuming.

Evaluate UID Construction

You might want to reevaluate how your site constructs the UID because you might want the fields of the UID to closely correspond to functions served by secondary authorization IDs. For example, a UID such as *ccdddfffllllllll* might represent the following types of information:

cc

City

ddd

Department

fff

Function

llllllll

User or logonid

You might also want users to be able to dynamically alter their UIDs. TSO or batch users that use their logonids as their primary IDs can dynamically alter the UID in CA ACF2 for DB2 when connecting to DB2. To do this, you must include the GROUP logonid field somewhere in the string. This feature makes the UID even more flexible and powerful by giving authorized users the privileges of another group not included in the UID. For example, when you sign on, you could enter ACCT as the group that you want to associate with. If you are authorized, you are able to access any resource that a person in the ACCT group can access. This is similar to using secondary IDs to supply additional privileges to users.

The UID is also crucial to using the CA ACF2 for DB2 conversion utility, which converts GRANT catalog entries to rule entries. If your UID concatenation does not include the logonid field, CA ACF2 for DB2 places the logonid in the UID field and creates a comment for the rule entry.

To find out more about the UID, see the following guides:

UID Information	Guide Reference
For general information	*CA ACF2 for z/OS Implementation Planning Guide* and *General Information Guide*
To modify the UID	*CA ACF2 for z/OS Installation Guide*
To dynamically alter the UID	*CA ACF2 for z/OS Administrator Guide* *CA ACF2 Option for DB2 Administrator Guide*

Evaluate Use of Exits

You can use CA ACF2 for DB2 exits to resolve site dependencies or provide transition paths. The DB2PRE and DB2POST exits let you adjust input to the rule interpreter before validation and adjust output from the CA ACF2 for DB2 rule interpreter after validation. To use these exits, see the “Customization” chapter.

You might want to reevaluate your use of the IBM DSN3@SGN and DSN3@ATH exits that give users additional privileges through secondary IDs. The DSN3@SGN and DSN3@ATH exits associate secondary authorization IDs to primary authorization IDs. IBM supplies these as default and sample exits. CA ACF2 for DB2 also supplies exits that you can use in place of the IBM-supplied exits. See the “Using Secondary Authorization IDs,” appendix if you want to use these exits.

You might be able to use the UID to completely eliminate the use of secondary authorization IDs. If you rely on the UID instead of secondary IDs to determine a user’s privileges, you do not need to remove these exits. You should, however, remove any of the CA ACF2 source groups associated with secondary IDs. CA ACF2 for DB2 will use the exits to set the primary and current SQL IDs, but will not use them to set the secondary IDs. The exits will set a return code of 0 for these IDs. See the documentation in the source code to learn how these IDs are set.

Evaluate Who Should Get Access to What

CA ACF2 for DB2 rules grant access to all DB2 resources, such as databases, tables, and system privileges. Before you can write and implement CA ACF2 for DB2 rules, you must be a security administrator or one must grant you change authority through a control statement (%CHANGE or %RCHANGE) of the rule set.

To make your rules as effective as possible, answer these questions before you start writing the rules:

- What are the names of the resources that I want to share?
- Which users should be able to use system privileges and utilities?
- Which users should own the resources?
- Which users should be able to change the rules? Do I want to restrict any of these users?
- Who do I want to share resources with? What are the UIDs of the users I want to share the resources with?
- Can I group and mask them?
- Should some users be privileged? Should these users be scoped?
- Should I trace access to a DB2 object? Should I trace a particular user’s access to an object? Should I trace use of a system privilege or utility?

- How do I want others to use the data? Should they be restricted in any way (such as to a column or to a certain function)?
- Should I determine certain time periods (shifts) when users can access the data?
- What privileges does DB2 require to access these resources?
- What table (database, and other) functions should users be permitted to access?

After you answer these types of questions, you can begin to create rules. We recommend that you use the conversion utility to create a set of general rules. Then you can edit these rules to be more specific. The Create Rules section explains the conversion utility and some of the considerations when writing rules. The *Administrator Guide* describes the syntax and mechanics of rule sets, and gives more information about writing rules.

Implement

Create Logonids

Because CA ACF2 should be installed on your system, you already have created logonids for many different uses. To use CA ACF2 for DB2, you must create DB2 subsystem logonids and user logonids. You can also create user logonids that are authorized to administer security functions. You can limit these privileged logonids to certain areas of responsibility. Because CA ACF2 for DB2 uses CA ACF2 processing to create and maintain logonids, you can create the necessary logonids before you install CA ACF2 for DB2. See the CA ACF2 *for z/OS* Administrator Guide for more information about creating logonids.

DB2 Subsystem Logonids

A DB2 subsystem executes as a collection of started tasks. A START DB2 console command executes the following started tasks:

*xxxx*MSTR
*xxxx*DBM1
*xxxx*DIST (for DDF users only)

In these started tasks, the name of the DB2 subsystem is *xxxx*, as defined in the IEFSSN*xx* member in SYS1.PARMLIB. In addition, DB2 uses IMS Resource Lock Manager (IRLM) to manage the locking of DB2 resources. You specify the name of this started task during the DB2 install process.

If your site does not require started task control (that is, NOSTC is specified in the GSO OPTS record), you do not need to define logonids for these started tasks. However, if you have selected started task control (that is, STC is specified in the GSO OPTS record), you must insert a logonid for each of the above started task names. If no logonids are found that match the started task names, CA ACF2 uses the DFTSTC logonid (that is, the value specified in the DFTSTC field of the GSO OPTS record). However, we recommend that you do not use the DFTSTC logonid value because DB2 usually has unpredictable security requirements.

After you create the DB2 subsystem logonids, you must assign them certain privileges. You should assign the STC privilege because you want only started tasks to use these logonids. You might want to assign the most powerful logonid attribute, NON-CNCL, because it might be difficult to write required rules for DB2. By default, CA ACF2 validates rules every time a logonid attempts to access a data set. However, with DB2, new and existing data sets are dynamically allocated. NON-CNCL grants the started task access to all z/OS data sets. It ensures the DB2 subsystem never abends with an S913 security violation. In addition, it provides a log of each data set access the started task makes that a rule set does not allow. This helps you determine what data sets the DB2 subsystem must access.

Instead of NON-CNCL, however, you might want to assign the MAINT attribute to the DB2 subsystem logonids. MAINT enables a logonid to access resources without rule validation or logging as long as the assigned logonid is executed from a defined program and library. To use MAINT with DB2 started task logonids, define the program and library identified in the procedure executing the started task in the GSO MAINT record. This attribute gives you more control over the IDs that can use this privilege than the NON-CNCL attribute. For more information about these logonid fields, see the CA ACF2 *Security for z/OS Administrator Guide*.

The following INSERT subcommands are examples of typical logonid record definitions that might be used for the DB2 started tasks. You might also require a logonid for the CAIENF started task (see the CA Common Services documentation for more information about CAIENF). The following INSERT subcommands would give the MAINT and STC attributes to these logonids.

```
INSERT DSNMSTR NAME(DB2 SYSTEM SERV) STC MAINT
INSERT DSNDBM1 NAME(DB2 DATABASE SERV) STC MAINT
INSERT DSNDIST NAME(DB2 DIST DATA FAC) STC MAINT
INSERT IRLMPROC NAME(IMS RSRC LOCK MGR) STC MAINT
INSERT ENF NAME(ENF STARTED TASK) STC MAINT
```

User Logonids

CA ACF2 for DB2 validates access requests to DB2 resources based on user logonids. Define a logonid for each user of DB2 resources. This includes all secondary authorization IDs and IDs passed from other systems, such as CICS or IMS.

You must determine the logonid record attributes that you want to define for a user to access a DB2 subsystem. You must decide how powerful you want the user logonid to be. In other words, how much authority do you want to grant to a logonid? The access authority you give to a user logonid record depends on the user's job function. If you want to make the user a security administrator, you must assign the SECURITY privilege to his logonid record. SECURITY gives the user more access authority than other user logonid records. Through the use of scoping, you can limit the authority that a security administrator can use. Most users need only normal authority. You would not assign their logonids special access authorities such as NON-CNCL or SECURITY.

You must also define logonids for users who connect to DB2 from other facilities, such as CICS, IMS, TSO, or batch. For example, CICS can pass transaction IDs and terminal IDs to connect to DB2 from CICS. You must create a logonid record for each of these IDs. Because some of these facilities can pass different IDs, you must determine which IDs are being passed and create logonids for them. See the CA ACF2 Security for *z/OS Administrator Guide* to create these logonids.

CICS

The resource control table (RCT) in CICS contains the ID that is passed to DB2 for authorization. You specify this ID in the AUTH parameter of the RCT. CICS lets you specify three choices for this ID. We suggest that you specify AUTH=USERID, causing CICS to pass the user's eight-character CICS sign-on ID for the authorization check. By using this ID, you maintain individual accountability.

These are the IDs that CICS can send to DB2 through the RCT:

CICS IDs	Description
USERID	The eight-character CICS sign-on user ID (recommended)
SIGNID	The user-defined ID in TYPE=INIT of the RCT
USER/OPIDENT	The three-character CICS sign-on operator ID
TERM	The CICS terminal name
TXID	The CICS transaction ID
GROUP	The RACF group authorization ID in TYPE=ENTRY of the RCT
String	An eight-character user-defined ID

IMS

In an IMS message-driven region, if the original user was signed on, the ID used in the authorization check is the user's sign-on ID. If the original user was not signed on, the LTERM name is used in the authorization check. In non-message driven regions, the PSB name is used.

TSO

For TSO regions, the logonid is used in the authorization check.

Batch and Call Attachment Facilities (CAF)

In a batch environment or with the call attachment facility (CAF), the authorization check validates the USER parameter on the job statement or the /*LOGONID or //*LOGONID statement in the JCL.

Create Rules

This section explains:

- Which resources require rule sets
- Who can create rule sets
- When to write rule sets
- How to create rule sets

See the *Administrator Guide* for more information about writing rules.

Which Resources Require Rule Sets?

CA ACF2 for DB2 requires a rule set for any DB2 resource that a user accesses. Rule sets grant users the same privileges that native DB2 security grants with SQL statements. To determine all the privileges (that is, rule sets) a user must have to access a resource, see the descriptions in the IBM *DATABASE 2 Command* and Utility *Reference* and the IBM *DATABASE 2 SQL Reference*.

For example, to let a user create a table, the user must have one of the following sets of privileges:

- Specific individual privileges that include the following:
  - The CRETAB (CREATETAB) privilege on the database to create the table
  - To implicitly create the table space for the table, the CRETS (CREATETS) privilege on the database and privileges to use the default buffer pool and default storage group
  - To explicitly name a table space, the privilege to use the table space
- The DBADM, DBCTRL, or DBMAINT authority.
- The system DBADM (SYSDBADM) authority. This authority lets users create the table but not update or access it.
- The SYSCTRL authority. This authority lets users create the table but not update or access it.
- The SYSADM authority.

These are the same privileges that a user must have if native DB2 security was used to grant access to the resource.

You grant these privileges through rule sets. For example, to enable a user with the USR01 logonid to create a table on the TEST DB2 subsystem, one of the following sets of rules must grant access:

- If you do not grant USR01 the SYSADM, SYSCTRL, SYSDBADM, DBADM, DBCTRL, or DBMAINT authority, you must grant him specific privileges to create a table.

    - To grant the CRETAB privilege on the FINPAY database, add the following rule set:

        ```
        $KEY(FINPAY)
        $TYPE(DBS)
        $SYSID(TEST)
         UID(*****USR) SERVICE(CRETAB) ALLOW
        ```

        This rule set gives all users whose logonids begin with USR the CREATETAB privilege on the FINPAY database for the TEST DB2 subsystem.

    - To enable USR01 to implicitly create table spaces in the FINPAY database, update the previous rule set as follows. When tables are created, they are placed in a table space. Thus, the privilege to create the table space is required. Here is the updated rule set:

        ```
        $KEY(FINPAY)
        $TYPE(DBS)
        $SYSID(TEST)
         UID(*****USR) SERVICE(CRETAB,CRETS) ALLOW
        ```

        Other rule sets that are required to implicitly create table spaces might include the right to use the default buffer pool and the default storage group (if a specific buffer pool and storage group are not named by the CREATETAB command). These rule sets most likely were created during processing by the conversion utility. These rule sets give all users the right to use the default buffer pool and storage group:

        ```
        $KEY(BP0)              $KEY(SYSDEFLT)
        $TYPE(BPL)              $TYPE(STG)
        $SYSID(TEST)              $SYSID(TEST)
         UID(-) ALLOW             UID(-) ALLOW
        ```

    - To permit USR01 to explicitly name the table space, he must have one of the following: SYSADM, SYSCTRL, SYSDBADM, DBADM for the database, or the right to use the table space. Previous examples show how to grant the SYSADM, SYSCTRL, SYSDBADM, and DBADM authorities. This example shows how to create a rule set that grants all USR users the right to use the table space:

        ```
        $KEY(tablespace)
        $TYPE(TSP)
        $SYSID(TEST)
         UID(*****USR) ALLOW
        ```

We suggest you mask your rule sets so that they can apply to multiple resources. Remember that you must mask the resource name to the maximum number of characters for that type of resource. You can also mask the UID parameter so that rule entries can apply to multiple users (in this case, to any user whose logonid begins with USR). This reduces the number of rule entries that you must write and maintain. Your site should establish a naming convention for all its objects to reduce rule writing and maintenance. If a convention is established, existing rule sets can automatically cover resources created by a database administrator.

Another method that can reduce your rule-writing effort is resource grouping. With this feature, you can write one rule set to govern access to multiple resources. For more information about masking and resource grouping, see the sections entitled Can You Mask CA ACF2 for DB2 Rule Sets? and Can You Use Resource Grouping? in the “Writing CA ACF2 for DB2 Rules” chapter in the *Administrator Guide* .

- To grant USR01 the DBADM, DBCTRL, or DBMAINT authority on the FINPAY database, add a rule entry for one of these authorities to the database rule set. This example shows only the DBADM authority:

  ```
  $KEY(FINPAY)
  $TYPE(DBS)
  $SYSID(TEST)
   UID(*****USR01) SERVICE(DBADM) ALLOW
  ```

- You can give USR01 the system DBADM authority to create a table. Add a rule entry to the SYSDBADM rule set as follows:

  ```
  $KEY(SYSDBADM)
  $TYPE(SYS)
  $SYSID(TEST)
   UID(*****USR01) ALLOW
  ```

- You can give USR01 the SYSCTRL authority to create a table. Add a rule entry to the SYSCTRL rule set, as follows:

  ```
  $KEY(SYSCTRL)
  $TYPE(SYS)
  $SYSID(TEST)
   UID(*****USR01) ALLOW
  ```

- To give USR01 the SYSADM authority, add a rule entry to the SYSADM rule set, as follows:

  ```
  $KEY(SYSADM)
  $TYPE(SYS)
  $SYSID(TEST)
   UID(*****USR01) ALLOW UNTIL(12/31/05)
  ```

  You can limit the use of SYSADM by USR01 until December 31, 2005. You can also restrict use of resources to certain time periods (that is, shifts).

Who Can Create Rule Sets?

To create a rule set, you must be a security administrator (possess the SECURITY privilege in your logonid record). To change a rule set, you must be a security administrator or be granted change authority through a control statement (%CHANGE or %RCHANGE). In CA ACF2 for DB2, creating a resource does not automatically mean you own it and, therefore, can access it. To own a resource, a security administrator must first create a rule set and identify you as the owner. The $LIDOWNER or $UIDOWNER control statement defines the owner of a resource. Owners have the same access privileges to a resource as a DB2 owner, but they cannot grant privileges to access the resource. This means that you do not need to write a rule entry for an owner to access the resource, but the owner cannot create or change the rule set to grant others access to the resource. If you want them to be able to grant others access, owners must have the SECURITY attribute in their logonids or be identified by the %CHANGE or %RCHANGE control statement.

One-way to delegate rule-writing authority is for a security administrator to compile and store a rule set that contains only the $KEY, $TYPE, $SYSID, and %CHANGE control statements (that is, no rule entries). This skeleton rule set lets a user designated by the %CHANGE control statement refine the rule set without requiring you to write rule entries.

When Can You Create Rule Sets?

You can create rules before or after you install CA ACF2 for DB2. Before installation you can build rule sets in partitioned data set (PDS) members. Then, you can compile and store them after CA ACF2 for DB2 is installed. After installation you can create new rule sets using the ACF command, ISPF panels, or the conversion utility.

We suggest that you build rule sets in partitioned data set members and that you use the $MEMBER control statement in the rule set. This enables you to select the member names into which you can later decompile the rule sets. If you do not use the $MEMBER control statement, the member names are generated for you when you decompile the rule sets into a PDS. To generate the member names, CA ACF2 for DB2 uses the names of resources. Because the names of tables, views, and table spaces can be longer than eight characters, member names created from these resource names are invalid. Also, resource names that are masked are invalid member names.

To compensate for generating invalid member names, the value of the MEMBER parameter specified on the SET subcommand is used to create member names that are valid. CA ACF2 for DB2 uses the MEMBER parameter, adds one, and precedes it with an at-sign (@). These generated member names are less than desirable because they do not reflect the contents of the member and can be difficult to track. To avoid this situation use the $MEMBER control statement when you build the rule sets.

How Do You Create Rule Sets?

You can create rule sets through the ACF command, ISPF panels, or the conversion utility.

Using the ACF Command

To administer CA ACF2 for DB2 through TSO, enter the ACF command when the TSO READY message appears on your panel. After you do this, your panel looks like this:

```
 READY
acf
 ACF
```

Note: If your TSO profile is set to NOMODE, you will see a question mark (?).

You can also enter TSO ACF on the command line of most ISPF panels or enter ACF directly from the TSO command panel.

After you are in ACF mode, you must establish the *command setting* before you can execute any ACF subcommands. The SET subcommand establishes the command setting. To process CA ACF2 for DB2 rules, enter the following with the type code of the resource that you want to process:

```
SET DB2(typecode)
```

See the *Administrator Guide* to learn more about the ACF subcommands used under this setting and about using ISPF panels.

Using ISPF Panels

If your site uses the IBM Interactive System Productivity Facility (ISPF), you can use ISPF panels to process rules.

To access ISPF panels

1. After you log on to z/OS, enter the ISPF command from the TSO READY mode to bring up the ISPF primary selection menu.

   The CA ACF2 SPF Option Selection Menu appears

2. Type the appropriate selection code to display the Selection Menu.

3. To process CA ACF2 for DB2 information, select option D.

Running the Conversion Utility

If you use native DB2 security, you can simplify rule writing by using the conversion utility to create rule sets. This utility converts SQL GRANT entries in the DB2 catalog into CA ACF2 for DB2 rule sets. We recommend that you run the conversion utility to create your first sets of rules because these provide a base from which to start writing rules. Also, these first rule sets provide a good opportunity to determine what has been happening in your DB2 environment. For example, you can identify if certain key resources have been vulnerable by reviewing the rule entries that were generated. This is also a good time to review whether each user granted access to the resource should continue to have access. You can add parameters to the rule entries that limit access in ways that internal DB2 security cannot. For example, you can limit users to specific time periods or specify expiration dates for access.

Before you run the conversion utility, ensure that:

- You have installed CA ACF2 for DB2 and the conversion utility.
- You have the proper DB2 authority to run the conversion utility. You need a rule that gives you EXECUTE authority for the conversion utility application plan CADB2C*nn*, where *nn* is the DB2 release level.
- You have the proper authority to access the Logonid database. You will need SECURITY, ACCOUNT, or AUDIT. If you are scoped, your scope must include the logonids that you intend to convert.
- You use E-SGP and/or X-SGP source group records in conjunction with the exits to associate secondary authorization IDs to primary IDs. The conversion utility supports both E-SGP and X-SGP source group records. Nested E-SGP and X-SGP records are also supported. The conversion utility also supports nested X-SGP records that contain masked secondary authorization IDs. However, masking of primary IDs is not supported.

  Note: Do not use the EXPAND parameter if the X-SGP records contain masked primary IDs.
- You create logonids for each entry in a secondary ID’s source group. This preliminary step ensures that CA ACF2 for DB2 creates valid rule entries for these IDs instead of commented rule entries. This also reduces the time spent to review and modify the IDs.
- You delete logonids from the Logonid database for any secondary IDs that will no longer be used. This step ensures that CA ACF2 for DB2 **does not** create valid rule entries for IDs that you want to eliminate from your subsystem.

To create CA ACF2 for DB2 rule sets using the conversion utility, follow these steps:

1. Run the conversion utility job
2. Evaluate rule sets for applicability
3. Modify converted rules or write new rule entries
4. Compile and store rules.

Step 1: Run the Conversion Utility Job

The conversion utility is a job (CP12CNVT) in the CAI.ACF2DB2.CACPJCL data set. The CP12CNVT job has three steps:

1. ALLOCATE

   This step allocates the partitioned data set (PDS) in which the output of the conversion program is stored. The name of the PDS is CAI.CONVOUT.

2. UNLOAD

   This step uses batch TSO to execute the DSN command. Change the SYSTEM keyword of the DSN command to the name of the DB2 subsystem that you are converting to CA ACF2 for DB2 security. If this step fails because the CONVOUT file is too small, increase the SPACE allocation in the ALLOCATE step and restart the job.

   Note: To run this step, you must have the EXECUTE privilege for the conversion utility application plan CADB2C*nn*, where *nn* is the DB2 release number.

3. CONVERT

   This step reads the data from each of the PDS members created in the UNLOAD step and places CA ACF2 for DB2 rule sets in the ACFRULES member.

   You can change two parameters in this step: the SUBSYSID parameter and the EXPAND or NOEXPAND parameter.

   - Change the SUBSYSID parameter to the value you want placed in the $SYSID control statement of the CA ACF2 for DB2 rules. This might be the same as the DB2 subsystem name from the UNLOAD step. Specify this parameter as four characters. If your subsystem name is typically three characters, you can use an asterisk to mask the remaining character (for example, DB2*).

   - You also might want to use a totally masked subsystem name such as ****. See the *Administrator Guide* for more information about masking the SYS You can also specify the EXPAND or NOEXPAND parameter in this step. The EXPAND parameter eliminates the use of secondary IDs by creating a rule entry for each ID associated with a secondary ID. This ID (that is, a user's primary ID) is used to determine access and not the secondary ID.

     During conversion processing, CA ACF2 for DB2 takes each ID in the DB2 catalog and searches the Logonid database. If a matching logonid exists, CA ACF2 for DB2 assumes that the ID is not a secondary ID and creates a rule entry using the UID information stored in the Logonid database. If a matching ID cannot be found, CA ACF2 for DB2 assumes it is a secondary ID and searches the source group by that name. This source group identifies the IDs that can associate with the secondary ID. A rule entry for each ID is created in the source group record, using information about each ID's UID concatenation from the Logonid database.

If one of the IDs is not defined in the Logonid database or as a source group entry, a comment is created for the rule entry by placing an asterisk in column one of the rule entry. The logonid is placed instead of the UID in the UID parameter. For example, the following output is created when it finds the ACCTPAY secondary authorization ID:

```
*BEGIN SECONDARY AUTHID: ACCTPAY
 UID(TFINPAYCLKUSER01) SERVICE(SELECT) ALLOW
 UID(TFINPAYMGRUSER55) SERVICE(SELECT) ALLOW
*UID(USER57) SERVICE(SELECT) ALLOW
 UID(TFINPAYAVPUSER68) SERVICE(SELECT) ALLOW
*END OF SECONDARY AUTHID
```

This example assumes that an entry source group (E-SGP) record of ACCTPAY exists with USER01, USER55, USER57, and USER68 as entries. USER57's UID information is not expanded because CA ACF2 for DB2 did not find this ID in the Logonid database. So, CA ACF2 for DB2 created a comment and placed the logonid in the UID parameter. With the NOEXPAND option, CA ACF2 for DB2 does not check source groups to create rule entries. Instead, CA ACF2 for DB2 creates a comment for the rule entry if it cannot find the ID in the Logonid database.

If this step fails because the CONVOUT file is too small, increase the SPACE allocation in the ALLOCATE step and restart the job. The output of this step is placed in the ACFRULES member and is used as input to the ACFBATCH utility. We recommend that you evaluate these rules before compiling them using ACFBATCH.

Step 2: Evaluate Rule Sets for Applicability

After the GRANT statements have been converted, you should review each of these rule entries for performance reasons, to add any additional CA ACF2 for DB2 features, or to revise commented rule entries.

- **Improve Performance**—The conversion utility builds one rule set for each GRANT statement stored in the DB2 catalog. If the GRANT statement names a secondary authorization ID and you specify the EXPAND option, CA ACF2 for DB2 builds a rule entry for each logonid in the secondary ID's source group. You can condense many of these rule entries into fewer rule entries based upon common UID information.

  For example, if ten GRANT statements name ten payroll clerks, the conversion program might create the following output:

  ```
  $KEY(PROD.PAYROLL_TABLE)
  $TYPE(TBL)
  $SYSID(DSN)
   UID(FINPAYPAYCLK01) SERVICE(ALL) ALLOW
   UID(FINPAYPAYCLK02) SERVICE(ALL) ALLOW
   UID(FINPAYPAYCLK03) SERVICE(ALL) ALLOW
   UID(FINPAYPAYCLK04) SERVICE(ALL) ALLOW
   UID(FINPAYPAYCLK05) SERVICE(ALL) ALLOW
   UID(FINPAYPAYCLK06) SERVICE(ALL) ALLOW
   UID(FINPAYPAYCLK07) SERVICE(ALL) ALLOW
   UID(FINPAYPAYCLK08) SERVICE(ALL) ALLOW
   UID(FINPAYPAYCLK09) SERVICE(ALL) ALLOW
   UID(FINPAYPAYCLK10) SERVICE(ALL) ALLOW
  ```

  You can consolidate these ten rule entries into one rule entry, as follows:

  ```
  $KEY(PROD.PAYROLL_TABLE)
  $TYPE(TBL)
  $SYSID(DSN)
   UID(FINPAYCLK) SERVICE(ALL) ALLOW
  ```

  In addition, your DB2 catalog tables might contain GRANT entries that are obsolete. The conversion utility will generate a rule entry for these entries. You should eliminate obsolete rule entries so that CA ACF2 for DB2 does not have to search through these to find the correct match.

- **Add DB2 Parameters**—You can add the following additional CA ACF2 for DB2 features to the basic rule entries. Review each utility-generated rule entry to determine whether to add these parameters to it.

Parameter	Description
$USERDATA	Passes information for user exits or documentation purposes
$LIDOWNER or $UIDOWNER	Establishes ownership of the DB2 resource

Parameter	Description
$MEMBER	Specifies the member name to be used for a decompile into a partitioned data set (PDS) if one is not provided with the decompile request
$MODE	Implements RULE mode and phases in the rule set
$OWNER	Contains the contents of the $OWNER control statement
$PREFIX	Specify in a resource rule to provide additional or alternate matching criteria for the resource name being validated
%CHANGE and %RCHANGE	Delegates rule writing authority
COLUMN	Specifies the columns of a table that a user can update.
NEXTKEY	Specifies the key of an alternate rule set CA ACF2 for DB2 checks if access to this resource is denied based on this rule entry.
NORULELNG	Overrides the use of the rulelong compiler when RULELONG is active.
$NOSORT	Prevents CA ACF2 for DB2 from sorting rules in a rules set when stored. The rules remain in the order they were first entered into the compiler through the terminal or a partitioned data set.
DATA	Passes information used in user exits or for documentation purposes.
LOG	Allows but logs an access attempt
SHIFT	Adds shift control
UNTIL or FOR	Adds expiration dates to the rule entries

Parameters such as SHIFT, UNTIL, FOR, and LOG enable you to place restrictions on the use of a resource. You can use them to audit access to resources by determining if access requests are made outside the limits. See Audit DB2 Resources, later in this chapter for more information.

- **Review Commented Rule Entries.** During the conversion process, when EXPAND is specified, commented rule lines are created whenever the conversion utility cannot find an ID or source group. CA ACF2 for DB2 checks the ID specified in the DB2 catalogs against the Logonid database. If the ID is not found, CA ACF2 for DB2 creates a comment for this rule entry. The logonid instead of the UID is then placed in the UID parameter of the rule entry.

  You should review each of these entries to determine whether you want to modify it. If the ID is a valid ID and you want to continue using it, you must add the ID to the Logonid database, update the UID parameter of the rule entry, and delete the asterisk that prefixes the rule entry.

Step 3: Modify Converted Rules or Write New Rules

To modify converted rule sets or write new rule sets to protect newly created resources, see the *Administrator Guide*.

Step 4: Compile and Store Rules

After you evaluate and modify the converted rules or create new rules, you must compile and store the rule sets in the Infostorage database.

- To compile the converted rules, execute the CP12ACFB job.
- To compile the new rule sets, issue the COMPILE and STORE subcommands. See the *Administrator Guide* for more information about using the COMPILE and STORE subcommands with the new rule sets. See the CA ACF2 *Security for z/OS Reports and Utilities Guide* for information about using the ACFBATCH utility.

Test CA ACF2 for DB2

You should perform the first test with CA ACF2 for DB2 in the system in a controlled environment. When you perform the test, you will want to include the security administrator, the database administrator, and the systems programmer responsible for the CA ACF2 installation.

In addition to establishing logonids, DB2 records, and rules, you must test several other items as follows soon after you install CA ACF2 for DB2 to ensure that they are functioning correctly. You also might want to perform additional tests to check local modifications or special processes.

- Test CAIENF startup. If CAIENF is already started because another CA Technologies product installed it, a message will tell you that it is already running. Be sure the CAIENF you are running is at the proper maintenance level to support CA ACF2 for DB2. Contact your local CA Technologies Support Center for the current maintenance level.
- Test DB2 startup. When you start your DB2 subsystem, you should receive a message telling you whether the DB2 subsystem is protected:

  DB2 SUBSYSTEM *xxxx* NOW PROTECTED BY EXTERNAL SECURITY

Test all local exits and modifications.

- Use the ACF SHOW subcommands to verify that the correct options for DB2 records, Global System Options (GSO) records, and the CA ACF2 Field Definition Record (ACFFDR) are active (for example, enter SHOW DB2, SHOW FIELDS, SHOW STATE, or SHOW SYSTEM. Or you can enter SHOW ACF2 to display everything that these subcommands can display.).
- Create the CA ACF2 for DB2 reports to test the jobs and produce reports that you can use to check other processing.

- Test the various CA ACF2 commands and subcommands that you will use to create and change DB2 records and CA ACF2 for DB2 rules. Ensure that they are working as expected and use them to help test and display rule-related options.
- Test the interfaces with other products (for example, CICS, IMS, and so on) by executing a few applications.

Migrate to Security

CA ACF2 for DB2 gives you various options to selectively migrate to full CA ACF2 for DB2 security. Although you cannot run CA ACF2 for DB2 and native DB2 security simultaneously on one subsystem, you can phase in security by:

- Subsystem
- Type of resource
- Specific resources

After you migrate to full CA ACF2 for DB2 security, the grants in the DB2 catalog are no longer used for securing the DB2 subsystem. If no other product uses the grant information, you can delete it to reclaim space in the catalog. Consult your DB2 database administrator before doing this cleanup to determine any impact that this might cause. For example, revoking a privilege that was used to create a view before CA ACF2 for DB2 is active can cause that view to be dropped and revoking a privilege required when binding a plan can cause the plan to be marked invalid.

If you have other products that use the grant information in the catalog (such as the IBM QMF), you might need to keep some of the existing grant data. Some of these products only need grant data related to tables. Then you can delete the non-table grant information. You can also replace the current grant data with grants that give access to PUBLIC. This does not create a security exposure since CA ACF2 for DB2 never uses the grant information in the catalog.

If you decide to keep duplicate grant information in the catalog for non-security product use (such as QMF), then you have to keep the grant information up-to-date with the CA ACF2 for DB2 rules. You can use the Catalog Synchronization Utility for this purpose.

Using the DB2 OPTS Record

This section explains the DB2 OPTS record and how you can use it to migrate and set global system options. See the *Administrator Guide* for more information about this record.

To selectively secure each DB2 subsystem, activate the DB2 OPTS infostorage record. This record determines whether CA ACF2 for DB2 or native DB2 security handles authorization checks. Specify ACTIVE to tell CA ACF2 for DB2 (that is, CAIENF) to intercept all DB2 security events. This is the default when you create a new DB2 OPTS record. ACTIVE lets you migrate the protection of each type of resource according to your site's needs. See the following section to learn about migrating resource types.

To let DB2 handle the authorization checks of a subsystem while you are setting up the subsystem's CA ACF2 for DB2 options, you can deactivate the subsystem's CA ACF2 for DB2 security. To do this, specify NOACTIVE when you create or change the DB2 OPTS record for that subsystem. In this way, DB2 continues to check authorization instead of CA ACF2 for DB2 checking authorization. The following subcommands show how to deactivate the DB2 OPTS record for the TEST DB2 subsystem.

```
READY
acf
 ACF
set c(db2)
 CONTROL
change opts noactive sysid(test)
```

To make these records effective, you must restart the DB2 subsystem that these records apply to.

The GSYSID field can be specified so that multiple DB2 subsystems can use the same resource rules, even if the subsystem SYSIDS cannot be masked to a common value. The value of the group SYSID will be substituted as the SYSID to match to resource rules when a validation is done.

The following example shows you how to create a DB2 OPTS record for the PROD DB2 subsystem and set the modes for all types of resources to LOG:

```
PROD / OPTS LAST CHANGED BY HUMMA03 ON 05/27/03-09:22
        ACTIVE BPLMODE(LOG) CONMODE(LOG) DBSMODE(LOG) FNCMODE(LOG)
        GSYSID() JARMODE(LOG) PLNMODE(LOG)
        PRCMODE(LOG) ROLMODE(LOGO) SCHMODE(LOG) SEQMODE(LOG) STGMODE(LOG)
        SYSMODE(LOG) TBLMODE(LOG) TSPMODE(LOG)
        TYPMODE(LOG)
```

Phasing in Resource Types

If you activate CA ACF2 for DB2 security for a subsystem, you can phase in protection by resource type including:

- Application plan, package, or collection
- Buffer pool
- Database
- Distinct type
- Function
- JAR file
- Role
- Schema
- Sequences
- Storage group
- Stored procedure
- System privilege or utility
- Table or view
- Table space
- Trusted context

Setting the Level of Protection

For each type of resource, you can set a different level (or mode) of protection using the DB2 OPTS record.

Note: Recognize that the mode you set for resources other than the one you are accessing can affect the access decision that CA ACF2 for DB2 makes. For example, a request to alter a tablespace can involve the DBSMODE and SYSMODE fields if other rule sets or the TSPMODE field do not grant access. This is because DB2 requires you to have ownership of the table space, DBADM on the database, or SYSADM, SYSCTRL, or SYSDBADM. For almost all requests, CA ACF2 for DB2 ultimately checks the system privilege mode if all other rule sets and resource modes deny access because SYSADM is the last authority that CA ACF2 for DB2 checks for. This means that SYSMODE is the final judge of whether access to a resource is permitted. Therefore, you should use caution when you set SYSMODE and others to QUIET or LOG.

QUIET Mode

QUIET mode allows the administrator to write rules and store them in the Infostorage database without CA ACF2 for DB2 using them. CA ACF2 for DB2 continues to check the logonid performing the function for the appropriate shift, scope, and so on, but does not validate the rules. No violation records are created.

LOG Mode

LOG mode lets CA ACF2 for DB2 make all the decisions concerning resource accesses, but does not deny any access to resources. Instead, CA ACF2 for DB2 creates SMF records of the violations. CA ACF2 reports show user accesses that would have been denied if you had chosen ABORT mode for the DB2 OPTS record or the rule set denied access or did not exist.

Log mode is typically used to start CA ACF2 for DB2 for a particular subsystem so that you do not need to write rules. In LOG mode an SMF record is written for every unauthorized access. Your reports will contain numerous entries until you write rules. Therefore, you will want to create rules with the conversion utility or write a few general rules for commonly used resources. This enables you to review the reports more easily.

The most permissive rule set that you can write for a resource is a one-rule entry that applies to all resources of that type on all DB2 subsystems. It also permits all access levels by anyone under any conditions. For example, you can enter a rule set for all tables with an authorization ID of USER01:

```
$KEY(USER01.********************)
$TYPE(TBL)
$SYSID(****)
UID(-) SERVICE(ALL) ALLOW
```

A few general rule sets will greatly reduce the volume of the reports immediately. You should review these general rules later and refine them to be more specific. When you are ready to write more specific rule sets, remove the temporary rule set. This causes all accesses to be logged again; because no rule applies to resource types, CA ACF2 logs all accesses as potential violations. Similarly, you can change the access permission in the existing rule entry from ALLOW to LOG for the same effect. You can then use the resulting CA ACF2 for DB2 reports to help write specific rules as appropriate. You should also consult with the data owners to determine if the remaining accesses are legitimate.

ABORT Mode

Put the resource type into ABORT mode to provide default protection. In this mode, CA ACF2 for DB2 checks all rules, logs all resource violations, and denies all violations. ABORT is the default value when the DB2 OPTS record is created.

RULE Mode

You can select RULE mode to migrate individual rule sets into LOG or ABORT mode. This mode lets you protect a specific resource at a different mode from the mode specified for the resource type. It lets you gradually store CA ACF2 for DB2 rule sets while still protecting the resource type at a certain level. See the next section to learn about phasing in rule sets.

Phasing in Specific Resources

You can phase in protection for specific resources by using RULE mode to migrate rule sets. RULE mode provides a flexible method for selectively converting to full CA ACF2 for DB2 security based on resource names. Using RULE mode, you can write CA ACF2 for DB2 rules, put them in production, test them, and refine them as appropriate. To use RULE mode, you

- Set the value of the *xxx*MODE field (that is, DBSMODE, TBLMODE, STGMODE, and so on) of the DB2 OPTS record for that resource type to RULE,*norule*,*no$mode*.
- Specify ABORT, LOG, or QUIET for the $MODE control statement in the CA ACF2 for DB2 rule set. (If you are migrating to full security, you will want to start with LOG or QUIET.)

When in RULE mode, CA ACF2 for DB2 bases its decision on the value of the $MODE control statement in the rule set. If it does not find a rule set, CA ACF2 for DB2 bases its decision on the default mode specified by the *norule* parameter in the DB2 OPTS record. If the rule set does not contain a $MODE control statement, CA ACF2 for DB2 bases its decision on the default mode specified by the *no$mode* parameter. The default mode can be one of three values: QUIET, LOG, or ABORT.

After you have finished writing rules, you can place the entire rule set in ABORT mode by removing the $MODE control statement (if the *no$mode* parameter specifies ABORT) or by changing the $MODE control statement to ABORT. Or you can leave RULE mode and place all resources under full CA ACF2 for DB2 security by changing the value of each *xxx*MODE field of the DB2 OPTS record to ABORT. The $MODE value in a rule set has no effect on processing if the *xxx*MODE field value of the DB2 OPTS record is set to anything other than RULE.

For example, suppose that the TBLMODE field of the DB2 OPTS record has the following values:

```
TBLMODE(RULE,ABORT,LOG)
```

Also suppose the following CA ACF2 for DB2 rule set protects a table resource named PERS.EMPLOYEE_PAY in the TEST DB2 subsystem. (UIDs at this site consist of the logonid preceded by three characters.)

```
$KEY(PERS.EMPLOYEE_PAY) TYPE(TBL) SYSID(TEST)
$MODE(LOG)
 UID(***USER01) SERVICE(SELECT) ALLOW
 UID(***USER02) SERVICE(UPDATE) PREVENT
```

USER02 wants to update the PERS.EMPLOYEE_PAY table. Although the applicable rule entry prevents USER02 from updating the table, the $MODE(LOG) control statement permits the access but logs it. CA ACF2 for DB2 checks this statement because the DB2 OPTS record for that resource (TBLMODE) specified RULE mode. If $MODE(QUIET) were specified, access would be permitted and not logged. Similarly, the access would be prevented if $MODE(ABORT) were set.

If the rule set did not exist, CA ACF2 for DB2 would use the mode specified for the *norule* parameter (that is, ABORT) in the DB2 OPTS record. If the $MODE control statement were absent from the rule set, CA ACF2 for DB2 would use the mode specified for the *no$mode* parameter (that is, LOG).

CA ACF2 for DB2 also checks other rule sets before access is ultimately granted. This means that the mode you set for these resources can override the access recommendation of a rule set. In the example above, CA ACF2 for DB2 would next check the database rule set for the DBADM privilege. If a matching rule set cannot be located or prevents access, CA ACF2 for DB2 then checks the DBSMODE field. If this mode specifies ABORT, CA ACF2 for DB2 goes on to interpret the SYSADM rule set. If neither of these authorities are granted, CA ACF2 for DB2 uses the SYSMODE field as the final judge of whether access is granted. This is because SYSADM is the last authority for which CA ACF2 for DB2 checks.

Protect DB2 System Data Sets

At this point, you will have already installed DB2, CA ACF2, and CA ACF2 for DB2. As part of your normal CA ACF2 security review, you should ensure that you have written access rules to protect all DB2 system data sets including:

- Databases
- Recovery logs
- Bootstrap data sets
- Distribution, target, SMP/E, and other installation data sets.

Check with your DBA or systems programmer for the names of these data sets.

To check whether access rules have been written for these data sets, use the DECOMP or LIST subcommand. For example, enter the following if the high-level index of the data set name is DSNC210, you enter:

```
DECOMP DSNC210
```

CA ACF2 for DB2 displays the rule entries that control access to the DSNC210 data sets.

Audit DB2 Resources

Unlike internal DB2 security, you can conduct a comprehensive review of DB2 resources without the help of your operations staff or database administrators (DBAs). CA ACF2 for DB2 lets you control what you want to audit, so the control over security is kept in the hands of security administrators. If you want to change what you are auditing, you can change it without asking an operator or someone with the proper authority to issue the START TRACE command. In addition, good business practices require that you know when accesses are denied and when changes to the rules and records that control access are made. CA ACF2 for DB2 automatically creates SMF records for denied access attempts and changes to CA ACF2 for DB2 rules and DB2 records. You do not have to start these traces in CA ACF2 for DB2. CA ACF2 for DB2 also does not impact your current use of DB2 audit trace facilities. You can continue to use these facilities with CA ACF2 for DB2.

With CA ACF2 for DB2, you can audit and report the following:

- Changes made to CA ACF2 for DB2 rules and DB2 records
- Access attempts that are denied by CA ACF2 for DB2
- Accesses to resources (including system privileges and utilities) that you want to record
- Accesses by users that you want to record
- All the resources that specific users can access
- All the users who can access specific resources

You can audit your DB2 resources with these reports described in the following subsections.

ACFRPTEL Report

The ACFRPTEL report logs all changes made to DB2 records and CA ACF2 for DB2 rule sets. Only users with the SECURITY privilege or users designated by the %CHANGE or %RCHANGE control statement can change rule sets. For example, you will want to know if a user changes the $MODE control statement. If $MODE is changed to QUIET (and you are in RULE mode), accesses to these resources are neither prevented nor logged regardless of whether the rule would normally allow or deny access. Also, because exits can bypass CA ACF2 for DB2 processing, you will want to know if someone defined an exit in place of a site-defined exit. This report lets you verify these changes. It is similar to the information provided by the class 2 audit trace records in DB2.

ACFRPTRV Report

The ACFRPTRV report provides information about all access attempts made to DB2 resources. It can tell you whether an access was denied because it violated a rule, an access was logged because you wanted to log each access to the resource, or an access was logged because you wanted to trace a user’s access to any resource.

- Denied access attempts are automatically logged.
- You can automatically log all accesses by specifying LOG on the rule entry. CA ACF2 for DB2 then allows access but creates an SMF record each time a user who matches the UID field requests the resource. This includes the use of system privileges and utilities, such as SYSADM, SYSOPR, RECOVER, and TRACE.

You can use the SHIFT and LOG parameters together to audit when accesses are being made to specific resources. If you suspect that accesses to certain resources are being made after work hours, you can create a SHIFT record that specifies your normal work hours and place it in the rule entry with the LOG access permission. If someone accesses the resource outside the normal hours, CA ACF2 for DB2 logs the attempt but allows it. In this way, you can survey when users are accessing resources.

Note: If you create an application plan from static SQL statements and validate the user's privileges at bind time, CA ACF2 for DB2 reports all logging requests for the resource that the plan refers to when the plan is bound. For example, if you specify that you want to log all update accesses to the EMPLOYEE.123 table and you are using static SQL statements to update the table, the ACFRPTRV report will record the update access when the plan is bound, not when the table is actually updated. This is true whenever DB2 validates a user's privileges at bind time. However, if you specify the RUN value for the VALIDATE option and DB2 validates the user's privileges at execution time, you will receive a report entry when the table is updated. DB2 does not always validate at execution time with this option. If the user has the required privileges at bind time, DB2 does not check his privileges at execution time.

For dynamic SQL statements, CA ACF2 for DB2 records the access at the time it is performed.

- The ACFRPTRV report also reports the access a specific user makes to any resource. To do this, you can update the user's logonid to include the TRACE attribute. When you specify TRACE, CA ACF2 for DB2 generates more than one SMF record if the user's accesses are violations or are logged. If you use TRACE on a logonid, remember that CA ACF2 for DB2 records the access when the logonid is validated. This means that CA ACF2 for DB2 creates the SMF record when a plan is bound, not when the resource is accessed. See the note above.

  You can also trace a user's logonid for diagnostic purposes. To determine why a user is unable to access a resource, set TRACE in his logonid. ACFRPTRV will generate the trace records for all accesses, including violations. You can then determine which rules you must write to grant access.

See the “Using Reports” chapter in the *Administrator Guide*, for detailed information about unusual loggings and violations in the ACFRPTRV report.

ACFRPTRX Report

The ACFRPTRX report provides information about which resources specific users can access. When you enter specific users' logonids as search criteria, CA ACF2 for DB2 reports the rule entries of each rule set that apply to each user. This report also tells you whether users:

- Are security administrators (their logonids specify SECURITY) or have NON-CNCL in their logonids
- Are resource owners (their UIDs match the $UIDOWNER or $LIDOWNER control statement)
- Can change the rule set (their UIDs match the %CHANGE or %RCHANGE control statement).

This provides important information about users' resource capabilities. Do they have limited access to the resource because of a rule entry or can they access the resource as owners? Can they change the rules protecting the resources and perhaps give others access?

ACFRPTXR Report

The ACFRPTXR report provides cross-reference information similar to the ACFRPTRX report. It gives you the names of users who can access a specific resource. When you enter a specific resource as search criteria, CA ACF2 for DB2 creates a list of applicable rules and displays the logonids of users whose user identification strings (UIDs) match the UID parameter in the rule entries. Auditors use this report to determine which users have access to a specific resource. It also gives you information about who can access the resource without rules (logonids that specify NON-CNCL or SECURITY) and which users are owners of the resource.

Chapter 3: Customization

This section contains the following topics:

Improving Performance

To improve performance of you can use resident directories or resident rule sets. *Directories* are lists of infostorage record keys that point to CA ACF2 for DB2 rule sets in common storage or on a direct access storage device (DASD). The main function of rule directories is to make CA ACF2 for DB2 rule sets readily available to CA ACF2 for DB2 systems by reducing the amount of I/O processing. The use of directories lets you copy the rule sets from the Infostorage database and keep them resident in global or local storage. The use of rule directories also lets you use CA ACF2 masking with CA ACF2 for DB2 rule sets.

Using the GSO INFODIR Record

By using the GSO INFODIR record, you can build globally resident directories. These directories are accessible to any DB2 subsystem because they are stored in the extended common storage area (ECSA). Use of globally resident directories increases performance because CA ACF2 for DB2 only has to build the directory once instead of building one for each address space.

- The GSO INFODIR record also lets you make the rule sets associated with the directories locally resident, globally resident, or transient.
- When rule sets are locally resident, CA ACF2 for DB2 loads them in an address space as needed during a validation request and keeps them in storage. This improves performance but sacrifices private storage. When rule sets are globally resident, CA ACF2 for DB2 loads the rule sets into global storage at directory-build time. This gives the highest performance but uses the most CSA storage. The last option is that CA ACF2 for DB2 can load the rule sets as needed and release them after they are used (that is, transient rule sets). If your rule sets are resident, you must rebuild the directory for it to reference rule sets that you have added or changed since you first built the directory.

Note: To use the long resource names possible in DB2 release 8.1 and above, you must use globally resident directories and globally resident resource rules. For more information on globally resident directories and resource rules, see “Using the GSO INFODIR Record” *in the Administrat*or Guide. For information about other requirements for long resource names, see the Insert Application Definitions step in the “Installation” chapter.

Using Masking

Rule directories let you use masking with rule sets. Masking lets you create fewer rule sets for multiple resources. This means that CA ACF2 for DB2 searches through fewer rule sets to find the matching one, which speeds processing.

Implementing Exits

If your site has security needs that are not part of the normal CA ACF2 for DB2 processing, you might be able to accommodate these needs by using the CA ACF2 for DB2 exits. An exit is a defined location where you can choose to receive control for your customized processing. You determine the functions that should occur at each exit and write a program to perform the processing.

Exits have major audit implications because they are permitted to alter the normal CA ACF2 for DB2 security processing. This section describes the CA ACF2 for DB2 exits that you can use to perform unique processing requirements.

Defining User Exits

Exits are defined by specifying the load module names in the appropriate field of the DB2 EXITS infostorage record. Before you invoke an exit program, you must create the DB2 EXITS record as follows:

```
INSERT EXITS DB2PRE(module) DB2POST(module) SYSID(sysid)
```

Each *module* value identifies the user-written exit program. The *sysid* identifies the DB2 subsystems that these exits apply to. You can obtain these names from your systems programmer.

You also must link any CA ACF2 for DB2 user exits into the CAILPA library with the RENT (reentrant), AMODE 31, RMODE ANY attributes and IPL with a CLPA or MLPA to activate them. CA ACF2 for DB2 obtains the addresses of these exits at initialization time. The user exits are given control in key 0, supervisor state, enabled, following standard z/OS linkage conventions. Because exit names are defined in the DB2 EXITS record, any module name is valid. However, you should avoid using exit names that begin with ACF.

Exits specified in the DB2 EXITS record but not found in the link pack area (LPA) are reported to the operator during CA ACF2 for DB2 initialization. Because user exits can alter the logic of CA ACF2 for DB2 validation, you should review exit design and implementation.

Exits are called using standard linkage as follows:

Register	Description
R1	Address of the ACGXITP parameter list as defined for the exit.
R13	Address of a standard register save area.
R14	Return address.
R15	Upon entry, this register contains the entry point address of the exit. Upon exit, this register contains a return code as defined for that exit.

Execution Environment

A step-must-complete enqueue might be in force when the exit is called. A reserve also might be outstanding with that enqueue if shared DASD is present.

The following sections describe the execution environment for each CA ACF2 for DB2 exit.

DB2 Prevalidation Exit (DB2PRE)

The DB2 Prevalidation exit (DB2PRE) is called at the beginning of each individual DB2 authorization check. Several such DB2 authorization checks can be made during the course of validating a single original DB2 request. For example, on the DROP database, you could see the DROP validation for the database, DBCTRL/DBADM for the database, then SYSDBADM, SYSCTRL, and SYSADM. This exit can force a decision (allow, allow but log, or prevent) for the validation request or modify the resource type or name to be used for validation. If the Prevalidation exit makes a decision, CA ACF2 for DB2 bypasses the Postvalidation exit.

Activating

To activate the Prevalidation exit, place the reentrant load module that contains the exit into the link pack area and specify the module name in the DB2PRE field of the DB2 EXITS record. If the specified module does not exist, CA ACF2 for DB2 initialization issues a warning message and continues. We recommend that you use a modified link pack area (MLPA) for testing. The CAI.ACF2DB2.CACPJCL data set contains a sample DB2 Prevalidation (SAMPPRE) exit.

Parameter List

The parameter list passed to the Prevalidation exit is named ACGXITP and is identical to the list passed to the Postvalidation exit. See the CA ACF2 *Security for z/OS Systems Programmer Guide* for field descriptions of ACGXITP.

The ACGXRB field of this list points to the ACGRSRC parameter block, which you can inspect in your exit. The ACFFLGS field in ACGRSRC identifies the type of function being requested (that is, the SERVICE parameter). To interpret this field, see the AD2DBEQU macro supplied with CA ACF2 for DB2. This macro explains how CA ACF2 for DB2 determines the value of this field. For an example of interpreting this field, see the SAMPPRE exit.

The CAIMAC library contains ACGXITP, ACGRSRC, and AD2DBEQU.

Register Conventions

On entry to the exit, standard register conventions are in effect. Register 1 contains the parameter list address, and register 0 contains the CA ACF2 for DB2 recommendation as follows:

DB2 Prevalidation Exit Execution	
Specification	DB2 EXITS record, DB2PRE field
Input	Standard calling sequence R1 Address of the ACGXITP parameter list

DB2 Prevalidation Exit Execution	
Output	R15 Return codes: 0 Continue normal processing 4 Allow request 8 Allow and log request 20 Prevent request

DB2 Postvalidation Exit (DB2POST)

The DB2 Postvalidation exit (DB2POST) is called after each individual DB2 authorization check is complete. Several such DB2 authorization checks can be made during the course of validating a single original DB2 request. For example, on the DROP database, you could see the DROP validation for the database, DBCTRL/DBADM for the database, then SYSDBADM, SYSCTRL, and SYSADM. CA ACF2 for DB2 passes its recommendation for the disposition of the request, which enables the exit to modify that disposition.

Activating

To activate the DB2 Postvalidation exit, place the reentrant load module that contains the exit into the system link pack area, and specify the module name in the DB2POST field of the DB2 EXITS record. If the module specified does not exist, a warning message is issued and initialization then continues. We recommend that you use a modified link pack area (MLPA) for testing. The CAI.ACF2DB2.CACPJCL data set contains a sample DB2 Postvalidation (SAMPPOST) exit.

Parameter List

The parameter list passed to the Postvalidation exit is named ACGXITP and is identical to the list passed to the Prevalidation exit. See the CA ACF2 *Security for z/OS System Programmers Guide* for field descriptions of ACGXITP.

The ACGXRB field of this list points to the ACGRSRC parameter block, which you can inspect in your exit.

The ACFFLGS field in ACGRSRC identifies the type of function being requested (that is, the SERVICE parameter). To interpret this field, see the AD2DBEQU macro supplied with CA ACF2 for DB2. This macro explains how CA ACF2 for DB2 determines the value of this field.

For an example of interpreting this field, see the SAMPPOST exit. The CAIMAC library contains ACGXITP, ACGRSRC, and AD2DBEQU.

On entry to the exit, standard register conventions are in effect. Register 1 contains the parameter list address, and register 0 contains the CA ACF2 for DB2 recommendation as follows:

DB2 Postvalidation Exit Execution	
Specification	DB2 EXITS record, DB2POST field
Input	Standard calling sequence R1 Address of the ACGXITP parameter list R0 Contains the CA ACF2 for DB2 recommendation as follows: 0 Allow request 4 Allow and log request 16 Prevent request
Output	R15 Return codes: 0 Continue normal processing 4 Allow request 8 Allow and log request 20 Prevent request **Note:** The return codes are four greater than the recommendation passed in register 0.

Using the DB2 Distributed Data Facility (DDF)

If you are a Distributed Data Facility (DDF) user, you can use CA ACF2 for DB2 to protect access to resources on your target DB2 subsystem. To validate DDF resource requests in the target system, ensure that the primary or secondary authorization ID of the request is defined in the CA ACF2 Logonid database of the target system.

DB2 calls the DSN3@ATH exit in the target system to provide secondary authorization IDs (if any). CA ACF2 for DB2 uses the logonid and any of its secondary IDs to perform validation in the target system. You must ensure that all these IDs are defined in the target system’s Logonid database.

Chapter 4: Troubleshooting

This section contains the following topics:

Diagnostic Procedures

See the following flowchart for a summary of the procedures to follow if you have a problem with a CA Technologies software product. Each of these procedures is detailed on the following pages.

☑

Collecting Diagnostic Data

In the following table, use the left column to categorize the problem your site has encountered. Then, follow the instructions in the corresponding right column to generate useful diagnostic data.

Type of Problem	Diagnostics
CAIENF	*CA Common Services Administrator Guide* and *Getting Started* CAS9DB database utility LIST command output Console messages CAIENF operator command output Output obtained from the ENF SVCDUMP or the ENF DUMP command Traces
CA ACF2 for DB2	ACFD2 messages Console messages SYSLOG for violation messages SVC dumps User dumps Output of ACF SHOW DB2 subcommand

Interpreting Diagnostic Data

When you have collected the specified diagnostic data, write down your answers to the following questions:

1. What was the sequence of events prior to the error condition?
2. What circumstances existed when the problem occurred and what action did you take?
3. Has this situation occurred before? What was different then?
4. Did the problem occur after a particular PTF was applied or after a new release of the software was installed?
5. Have you recently installed a new release of the operating system?

   Has the hardware configuration (tape drives, disk drives, and so forth) changed?

   From your response to these questions and the diagnostic data, try to identify the cause and resolve the problem.

If you are unable to resolve the problem, have the following information ready before contacting CA Technologies technical support:

- All the diagnostic information described in Collecting Diagnostic Data.
- Product name, release number, operating system, and genlevel.
- Product name and release number of any other software you suspect is involved.
- Release level and PUT level of the operating system.
- Your name, telephone number, and extension (if any).
- Your company name.
- Your site ID.
- A severity code. This is a number (from one to four) that you assign to the problem. Use the following to determine the severity of the problem:
  1. A “system down” or inoperative condition
  2. A suspected high-impact condition associated with the product
  3. A question concerning product performance or an intermittent low-impact condition associated with the product
  4. A question concerning general product use or implementation.

If you determine that the problem is a result of an error in a CA Technologies software product, you can use the online help system at SupportConnect.ca.com, to determine if a fix (APAR or PTF) or other solution to your problem has been published.

Accessing the Online Client Support Systems

To complement Technical Support, CA Technologies provides an online support system, SupportConnect. This system offers dial-up and online services to provide you with total service and support. To enroll in this service, contact your CA Client Service Representative.

Tracing CA ACF2 for DB2 Authorization Calls

When CA Technologies Technical Support instructs you to, you can execute the CADB2TRC program to turn the internal diagnostic tracing for CA ACF2 for DB2 on or off. The diagnostic trace messages provide CA Technologies Technical Support with information regarding the DB2 authorizations calls.

Following is sample JCL that you can use to execute CADB2TRC:

```
//jobname  JOB  acct.info,’DIAG TRACE’,CLASS=A,MSGCLASS=1
//CADB2TRC EXEC PGM=CADB2TRC,
//         PARM=‘TRACE=ON,TYPE=ALL,SUBSYS=DSNT’
```

PARM Keywords

You can execute CADB2TRC multiple times to activate multiple trace types. Each execution produces a list of all active trace requests. A request of TYPE=ALL,TRACE=OFF turns off all tracing requests.

The execute PARM for CADB2TRC indicates the DB2 threads for which CA ACF2 for DB2 produces diagnostic trace messages.

Valid PARM keywords are:

TRACE=ON|OFF|QUERY

This keyword is required.

- ON—Turns on diagnostic tracing.
- OFF—Turns off diagnostic tracing.
- QUERY—Produces a list of active traces.

TYPE=ALL|USER|JOB|CONNAME|CONTYPE

This keyword is required if you specify TRACE=ON|OFF.

- ALL—The trace is for all threads.
- USER—The trace is for a specific user or users.
- JOB—The trace is for a specific job or jobs.
- CONNAME—The trace is for a specific connection name or names.
- CONTYPE—The trace is for a specific connection type or types.

SUBSYS=*subsystemname*

This keyword is required. The value of subsystemname is the explicit name of the DB2 subsystem CA ACF2 for DB2 is protecting for which you are requesting diagnostic trace messages. You cannot mask subsystemname.

NAME=*typename*

This keyword is required when you specify TYPE=USER|JOB|CONNAME| CONTYPE. The value of typename is the name of the thread type to trace. You can specify the name with a mask as follows:

- ***** Matches any single character.
- - Matches all remaining characters, including null.

When you specify TYPE=USER, typename specifies the primary authorization ID to trace.

When you specify TYPE=JOB, typename specifies the name of a job to trace.

When you specify TYPE=CONNAME, typename specifies the connection name to trace (such as BATCH, TSO, DB2CALL). A connection name of SERVER is used for distributed calls when the remote system is not DB2.

When you specify TYPE=CONTYPE, typename specifies the connection type to trace (such as BATCH, DIST, MASS, SASS).

CADB2TRC Output

The output of the CADB2TRC program, showing the processing of the TRACE request from the execution PARM, is issued using the WTO (write to operator) service with a route code of 11. This route code causes the output to be written to the job log of the job executing the CADB2TRC program.

Diagnostic Trace Output

The diagnostic trace messages that are issued when you turn a trace on are issued using the WTO (write to operator) service with a route code of 11. This route code causes the diagnostic trace messages for a DB2 request to be written to the job log of the job that is executing the DB2 request.

Many installations define their z/OS consoles to receive all WTO messages, regardless of route code. If you have defined a z/OS console to receive WTO messages with a route code of 11, the diagnostic trace messages will also be sent to the z/OS console.

Important! If you have defined a z/OS console to receive WTO messages with a route code of 11, you should not turn on diagnostic traces that will generate large amounts of display output, for example, a global trace on a high volume DB2 subsystem. Turning on a high volume diagnostic trace may flood the z/OS console with messages or create a z/OS WTO buffer shortage.

Product Releases and Maintenance

CA ACF2 for DB2 provides new users with a distribution tape containing the current version of the system. Clients are requested to operate only under currently supported releases of CA ACF2 for DB2.

Standard user documentation is also provided to CA ACF2 for DB2 users. Updates to this documentation are provided automatically to all customers having current maintenance agreements. You can also access the latest documentation on SupportConnect.ca.com.

Customers with current maintenance agreements also receive ongoing CA ACF2 for DB2 maintenance. When a new release of the system is available, a notice is sent to all current CA ACF2 for DB2 customers.

Requesting Enhancements

CA Technologies welcomes your suggestions for product enhancements. All suggestions are considered and acknowledged. You can use either of two methods to request enhancements:

- Contact your Account Manager who will initiate a Demand Analysis Request (DAR) for you.
- Request this information on CA Technologies web site at www.ca.com.

Chapter 5: Using Secondary Authorization IDs

This section contains the following topics:

Using Exits

If you want to associate a secondary authorization ID to a primary authorization ID, you must use an exit. DB2 invokes one of two exits to let you inspect or modify the user's primary authorization ID. The *primary authorization ID* is usually the logonid that is accessing the system. Both exits can associate a list of secondary authorization IDs to a user's primary authorization ID.

The exits you use are DSN3@ATH and DSN3@SGN. You must link edit them into the DB2 SDSNEXIT library.

- DSN3@ATH is an authorization connection exit that gets control at CICS or IMS startup and during each TSO or batch job connection to DB2. It also gets control during distributed data facility (DDF) connections.
- DSN3@SGN is a sign-on connection exit similar in function to DSN3@ATH. DSN3@SGN gets control whenever a CICS or IMS user connects to DB2.

The following table details the exit used for each type of connection.

SAF Connection Type	Attach Exit (DSN3@ATH)	Sign-on Exit (DSN3@SGN)
BATCH	TSO Batch jobs Started tasks Utilities CAF jobs Anything other than CICS or IMS tasks	Does not apply
SASS (CICS)	CICS recovery coordinator task	CICS recovery coordinator task CICS transactions

SAF Connection Type	Attach Exit (DSN3@ATH)	Sign-on Exit (DSN3@SGN)
MASS (IMS	IMS control region (recovery coordinator)	IMS control region MPPs BMPs Fast path DL/I batch
DIST	Distributed data facility (DDF)	Does not apply

CA ACF2 also provides two sample exits that you can use instead of the IBM-supplied default exits. They can associate a list of secondary authorization IDs to a logonid or primary authorization ID. They can also set the value of the current SQL ID. These exits, CSECT DSN3@ATH and CSECT DSN3@SGN, reside in members ACF3@ATH and ACF3@SGN, respectively, of the CAI.CAIMAC library.

Defining Source Groups

To use the CA ACF2 exits at your site, you must define source group records to identify the secondary IDs. To determine the secondary ID, the exits provide the DB2 primary authorization ID to CA ACF2. CA ACF2 reads CA ACF2 source group records, prepares a list of secondary authorization IDs, and returns the list to DB2.

You can use cross-reference source group (X-SGP) or entry source group (E-SGP) records to identify secondary IDs. The conversion utility, which translates DB2 catalog entries into rule entries, does not support masking of the primary IDs (logonids) on X-SGP records. So, if you plan to use the conversion utility with the EXPAND parameter, you should not use masking on the primary IDs (logonids).

Each source group record defines the secondary authorization ID as the source group record name and the primary authorization IDs (that is, logonids) as entries in each source group. Therefore, each source group record contains the logonids or primary IDs that are authorized to associate with a secondary ID. When a logonid connects to DB2, CA ACF2 returns a list of all secondary IDs (that is, source group record names) that the ID is permitted to associate with.

The following examples show you how to define entry source group (E-SGP) records for secondary authorization IDs. In these examples, LID1, LID2, and LID3 are the logonids. The secondary authorization IDs are SECID1, SECID2, and SECID3.

This example shows how to create source group (E-SGP) entries for each secondary ID.

```
READY
acf
 ACF

set entry(sgp)
  ENTRY
insert secid1 type(sgp) newdata(lid1)
   TYPE: SGP   ENTRY: SECID1    1 DATA ITEM
     LID1
  ENTRY
insert secid2 type(sgp) newdata(lid1)
   TYPE: SGP   ENTRY: SECID2    1 DATA ITEM
     LID1
  ENTRY
insert secid3 type(sgp) newdata(lid1)
  TYPE: SGP   ENTRY: SECID3    1 DATA ITEM
    LID1
 ENTRY
```

The next example adds LID2 to the SECID1 source group and LID3 to the SECID1 and SECID3 source groups:

```
change secid1 newdata(lid2)
   TYPE: SGP   ENTRY: SECID2    2 DATA ITEMS
    LID1
    LID2
 ENTRY
change secid1 newdata(lid3)
   TYPE: SGP   ENTRY: SECID2    3 DATA ITEMS
    LID1
    LID2
    LID3
 ENTRY
change secid3 newdata(lid3)
  TYPE: SGP   ENTRY: SECID3    2 DATA ITEMS
    LID1
    LID3
End
```

The following table shows the list of secondary authorization IDs that CA ACF2 passes to DB2 when each logonid connects to DB2:

If the logonid is:	CA ACF2 passes this list:
LID1	SECID1 SECID2 SECID3
LID2	SECID2
LID3	SECID2 SECID3

Defining Logonids

After you connect to DB2 and associate with a secondary authorization ID, CA ACF2 for DB2 treats secondary IDs identical to logonids. When you use a privilege assigned to a secondary ID, CA ACF2 for DB2 checks the Logonid database to verify the secondary ID. Therefore, you must ensure that you define logonids for each secondary authorization ID that you use.

You can insert secondary IDs without the CA ACF2 privileges of JOB, TSO, CICS, IMS, or STC. In this way, users cannot use these IDs outside of DB2 for other purposes. Users can only use them in DB2. See the “Maintaining Logonid Records” chapter in the CA ACF2 *Security for z/OS* Administrator Guide for more information about creating logonids.

Chapter 6: Using SAF

This section contains the following topics:

Writing SAF Resource Rules

SAF routes security information between the application (DB2) and the z/OS security product (CA ACF2). It provides a standard, central interface for security requests. You can use it to validate access to DB2 when an address space first requests to use a DB2 resource. Because DB2 does not provide a user sign-on with password verification, SAF provides a way to control access to DB2 for subsystems without CA ACF2 for DB2.

For subsystems using CA ACF2 for DB2, the use of SAF might be unnecessary because you can use CA ACF2 for DB2 rule sets to control access to DB2 resources and use the $SYSID control statement to determine which DB2 subsystems a user uses to access a DB2 resource.

Use the CA ACF2 SAF interface to translate a SAF call into a CA ACF2 request. Then resource rules can validate the CA ACF2 request to determine whether access to the DB2 subsystem is granted.

To write resource rules to protect each DB2 subsystem, you must understand how the SAF call is written. The SAF call uses the RACROUTE macro and keywords:

```
RACROUTE
  REQUEST=AUTH
  CLASS=DSNR
  ENTITY=xxxx.yyyyy
```

The ENTITY value, *xxxx.yyyyy*, represents the z/OS subsystem name (*xxxx*) of the DB2 subsystem. The *yyyyy* variable identifies the type of system making the access request. This variable is one of the following choices:

System Types	Description
BATCH	A batch job or a TSO session.
MASS	A multiple address space system. IMS uses this qualifier.
SASS	A single address space system. CICS uses this qualifier.
DIST	A distributed data facility (DDF) session.

The SAF interface converts the SAF call into a resource validation call with the type code of SAF. Use the GSO CLASMAP record to change the default type code. The $KEY for the resource rule is formatted as *xxxx.yyyyy* where *xxxx.yyyyy* represents the ENTITY value described previously.

The following are sample ACF commands and a resource rule that permits the CICSRGN logonid to connect to the PROD DB2 subsystem from CICS.

```
  READY
acf
 ACF
set resource(saf)
  RESOURCE
compile
 ACF70010 ACF COMPILER ENTERED
 . $key(prod.sass)
 .  uid(******CICSRGN) allow
```

In reports, the infostorage record key for this resource rule appears as R-SAF-PROD.SASS.

Chapter 7: Checklists

This section contains the following topics:

Task Checklist

The planning checklist is used to ensure the smooth flow of installation and implementation events. You can also use it to ensure that nothing is overlooked when you install and implement CA ACF2 for DB2.

Step	**Task**
Step 1	Decide on centralized or decentralized security
Step 2	Appoint your implementation team.
Step 3	Plan and coordinate your implementation schedule
Step 4	Distribute documentation
Step 5	Provide adequate training.
Step 6	Review and tailor your security policy.
Step 7	Establish local naming conventions.
Step 8	Identify existing security mechanisms.
Step 9	Ensure uniqueness of user identifications.
Step 10	Review operating system configuration.
Step 11	Determine CA ACF2 control options.
Step 12	Determine CA ACF2 for DB2 options in DB2 records.
Step 13	Evaluate use of secondary authorization IDs.
Step 14	Evaluate UID construction.
Step 15	Evaluate use of exits.
Step 16	Evaluate who should get access to what.
Step 17	Ensure your system meets requirements.
Step 18	Ensure you have all the installation materials.
Step 19	Install CA ACF2 for DB2 (follow installation checklist that is provided).

Step	Task
Step 20	Create logonids.
Step 21	Create CA ACF2 for DB2 rules.
Step 22	Test CA ACF2 for DB2.
Step 23	Migrate to CA ACF2 for DB2 security: o Step 23a By subsystem o Step 23b By resource type o Step 23c By specific resource
Step 24	Protect DB2 system data sets
Step 25	Audit DB2 resources.
Step 26	Tune your system, if needed.
Step 27	Implement exits, if needed.

Installation Checklist

The following list summarizes the steps you must follow to install CA ACF2 for DB2. You should review this list before you install, and then use it as a checklist during the actual installation. Save any output you receive during installation to help resolve any problems that might arise.

Steps	Description
Step 1	Review System Requirements
Step 2	Load Installation Sample JCL Library
Step 3	Unload Online Documentation
Step 4	Optionally Remove Previous Product Release
Step 5	Allocate Distribution Librarires
Step 6	Allocate the CAI Software Target Libraries
Step 7	Allocate and Format the CAI SMP/E Data Sets
Step 8	RECEIVE Processing
Step 9	APPLY Processing
Step 10	ACCEPT Processing
Step 11	Insert Application Definitions (APPLDEFs)
Step 12	Update ISPF Procedures

Steps	Description
Step 13	Update CA ACF2 Main Panels
Step 14	Update CAIENF Load Library
Step 15	Define CA ACF2 for DB2 to CAIENF
Step 16	IPL the System
Step 17	Optional Link Edit of CADB2XAC Exit
Step 18	Insert DB2 OPTS Records
Step 19	Install Conversion and Synchronization Utilities