

Access To Data In The Banner Database: Rules, Processes, And Procedures

Written by Edward Harter

Approved by the Systems Users Group (SUG)

Russell Banta, VP, Finance

Doris Helmich, VP, Student Services

John Irvine, VP, Instructional Services

Linda Day, Curriculum Support Analyst

Steven Ward, Director HR Operations

Edward Harter, Chief Information Officer

James Adams, Database Administrator

Revision 0

02/01/08

Table Of Contents

Table Of Contents	i
1. PURPOSE AND SCOPE.....	1
2. LIST OF ACRONYMS	1
3. BACKGROUND INFORMATION	2
3.1. Overview Of Oracle Security.....	2
3.1.1. Database Privileges.....	2
3.1.2. Oracle Roles.....	3
3.2. Overview Of Banner Security.....	4
3.2.1. Banner and Oracle Forms	4
3.2.2. The Banner Security Scheme.....	4
3.3. Overview Of CASA Security.....	7
3.3.1. CASA and Banner Data	8
3.3.2. CASA Data Security	8
3.4. Overview of Network Security	11
3.4.1. Remote Access.....	12
3.4.2. Internal Access.....	13
3.4.3. Intrusion Detection System.....	13
4. ACCESS REQUESTS AND APPROVALS	14
4.1. Normal Request Processing	14
4.1.1. Internal Network Access.....	14
4.1.2. Remote Network Access.....	16
4.1.2. Banner Access Requests	16
4.1.3. CASA Access Requests	18
4.2. Special Request Processing.....	20
5. ONGOING MONITORING AND MAINTENANCE	20
6. SOFTWARE CHANGES	20
6.1. Changes to Banner	21
6.1.1. In-House Modifications	21
6.1.2. Vendor-Provided Patches and Upgrades	22
6.2. Changes to CASA	23
7. EMERGENCY SITUATIONS	24

1. PURPOSE AND SCOPE

The purpose of this document is to set forth the rules and procedures for controlling access to data in the Banner Enterprise Resource Planning (ERP) system at Central Arizona College (CAC). Inasmuch as this data is accessed and manipulated by various computer programs, the document also deals with rules both for controlling access to these programs by users and for controlling changes to these programs by technical staff.

The document is intended for use by for everyone who uses the CAC information systems and, especially, for anyone who requests access or changes to these systems.

The document contains frequent references to the *Systems Users Group*, or *SUG*. The SUG is a group of supervisory and management-level personnel who represent the major user and functional constituencies of the College: Finance, Human Resources (HR), Student Services, Instructional Services, Curriculum, and Information Resources and Services (IRS). As a cross-functional management-level group, the SUG plays a central role in the process of approving requests for access to Banner data because (i) the functionality of Banner components frequently crosses departmental and organizational boundaries and (ii) the data that is prepared and manipulated by Department A is frequently used by Department B. From a practical standpoint, the processes and procedures described in this document are owned by the SUG.

The document has the initial approval of the SUG and will be kept under strict configuration and change control in Visual Source Safe (VSS).

2. LIST OF ACRONYMS

ACL	Access Control List
BURF	Banner Upgrade Request Function
CAC	Central Arizona College
CARS	Central Arizona Request Services
CASA	Central Arizona Software Applications
DARF	Data Access Request Function
DBA	Database Administrator
DBMS	Database Management System
DDL	Data Definition Language
DML	Data Manipulation Language
ERP	Enterprise Resource Planning
HR	Human Resources
IDS	Intrusion Detection System
IRS	Information Resources And Services
LAN	Local Area Network
PL/SQL	Procedural Language/Structured Query Language
PSRF	Programming Services Request Function
SQL	Structured Query Language
SUG	Systems Users Group
UC	User Class
VLAN	Virtual LAN
VP	Vice President

VPN Virtual Private Network
VSS Visual Source Safe (Microsoft)
WAN Wide Area Network

3. BACKGROUND INFORMATION

All Banner data is stored in a database managed by the Oracle database management system (DBMS). A database managed by the Oracle DBMS is typically (and somewhat loosely) called an ‘Oracle database’ and will be so called throughout this document.

There are three major layers of security that physically control access to Banner data at CAC: network security, Banner security, and Oracle security. See Figure 1.

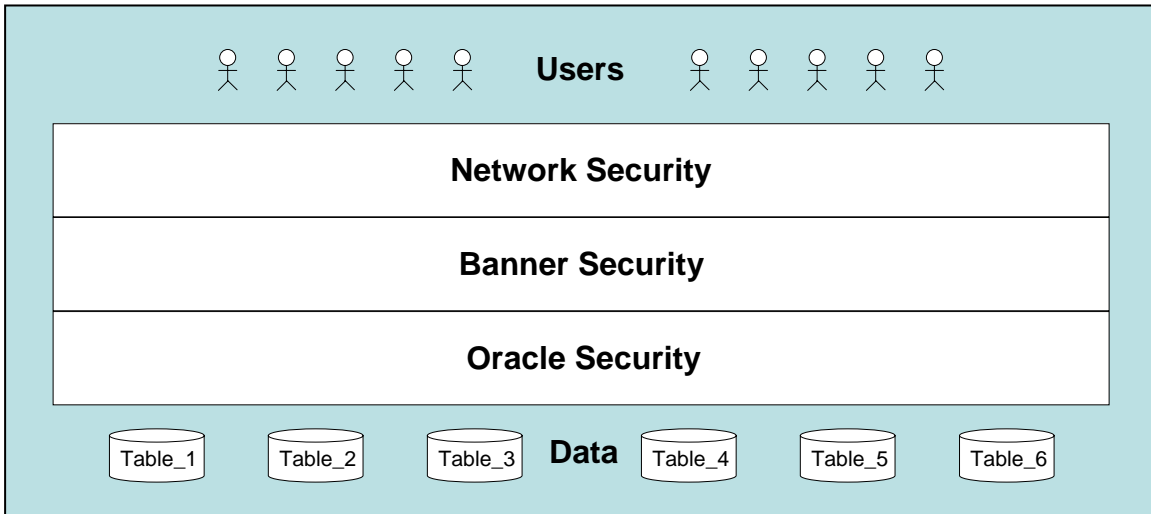


Figure 1: Major Layers of Physical Security

This section provides an overview of these layers from the bottom up because this is the simplest way to describe and understand them. Oracle security is described in Subsection 3.1, Banner security in Subsection 3.2, and network security in Subsection 3.3.

Subsequent sections of the document describe (i) how users are granted access through these layers; (ii) how they are granted access to computer programs that read and manipulate the data; and (iii) how CAC controls changes to these computer programs.

3.1. OVERVIEW OF ORACLE SECURITY

3.1.1. Database Privileges

With an Oracle database, a user cannot do anything unless he is explicitly granted the ‘privilege’ to do it. For example, if a user has not been granted the CONNECT privilege, he cannot even log in. In addition to the CONNECT privilege, the key privileges for most users are

- SELECT This gives a user the ability to select and view data from specified database tables.

- UPDATE This gives a user the ability to modify data values in specified database tables.

DELETE	This gives a user the ability to delete rows (records) from specified database tables.
INSERT	This gives a user the ability to add new rows (records) into specified database tables.

These privileges are granted through Oracle data definition language (DDL) commands. These commands can be issued in the following ways—either:

1. By the Database Administrator (DBA).
2. By the ‘owner’ of the database tables in question.
3. Indirectly, by the Banner application itself—as explained in *Section 3.2. Overview of Banner Security*.

For example, the DDL command `GRANT SELECT ON TABLE_A TO USER_B` gives `USER_B` the ability to select and view data from `TABLE_A`.

PROCEDURAL RULE: If a user needs one of these privileges, it must be granted either by the DBA or by the Banner application itself. If granted by the DBA, the grant must be issued with the approval of the SUG in response to a documented request. The approval process for granting database privileges is described in *Section 4.1.3. CASA Access Requests*.

3.1.2. Oracle Roles

Because several users can require the same set of privileges in order to do their jobs, these privileges are typically granted indirectly, through the intermediary of Oracle *roles*.

An Oracle role is a bundled set of privileges. For example, `ROLE_X` might consist of the following privileges:

```
SELECT ON TABLE_1
UPDATE ON TABLE_1
INSERT ON TABLE_1
```

With `ROLE_X` in place, the DBA can grant all of these privileges to `USER_I`, `USER_J`, and `USER_K` with a single DDL command, as follows:

```
GRANT ROLE_X TO USER_I, USER_J, USER_K
```

PROCEDURAL RULE: Except for the roles that are pre-created by the Banner system itself, roles may be created only by the DBA. If created by the DBA, the role must be created with the approval of the SUG in response to a documented request. The approval process for granting Oracle roles is also described in *Section 4.1.3. CASA Access Requests*.

When roles are employed, the Oracle security architecture consists of layers, as depicted in Figure 2 (next page). The double-headed arrows indicate that the relationships are many-to-many: A user can have multiple roles and a role can belong to multiple users; a role can provide access to multiple database tables; and a database table can be accessed through multiple roles.

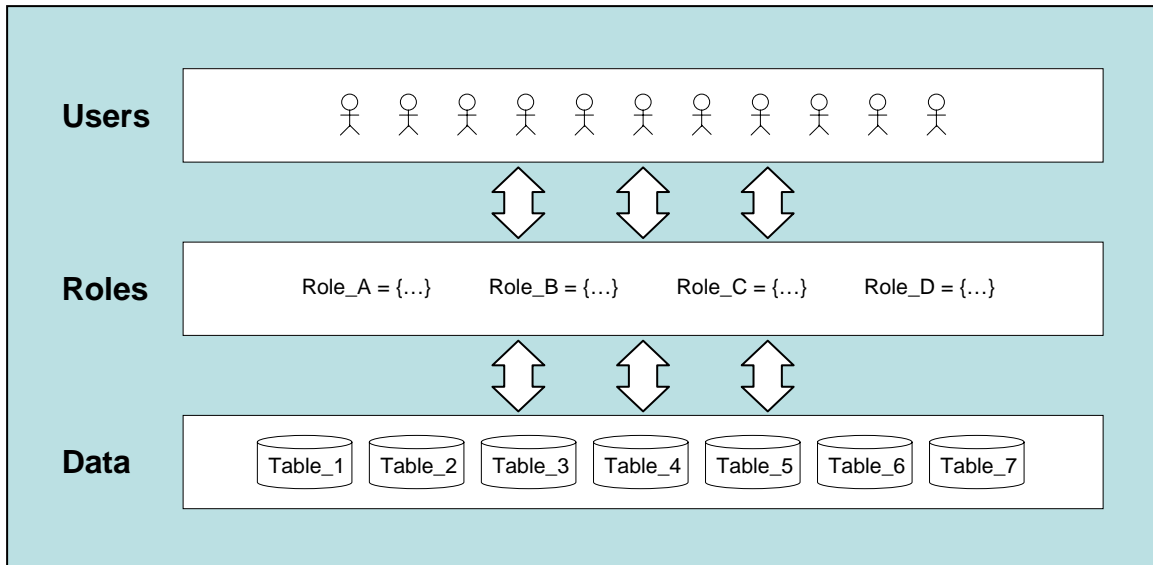


Figure 2: Oracle Database Security Schematic

Note that, as indicated in Section 3.1.1 above, it is possible for the DBA to grant specific database privileges to individual users *directly*, without using the intermediary of roles. However, in practice this can lead to database clutter and ultimately to visibility and control problems. As a consequence, this practice is discouraged.

PROCEDURAL RULE: The DBA can grant specific database privileges to individual users *directly*, but only in emergency situations—and only with an appropriate level of management control, as described in *Section 7. Emergency Situations*.

3.2. OVERVIEW OF BANNER SECURITY

3.2.1. Banner and Oracle Forms

Banner is a set of application programs that interact with an Oracle database. The programs are created with a toolset known as Oracle Forms. Oracle Forms is similar to Visual Basic and Visual C++, the primary difference being that Oracle Forms is designed specifically to interact with Oracle databases. When a user logs in to Banner, she logs into the Oracle database through an Oracle Forms program. Individual Oracle Forms programs are frequently referred to simply as *Oracle forms* (with a lowercase 'f').

An Oracle form consists of a graphical user interface with programming functionality built into it. The programming functionality is written in the language known as Oracle *Procedural Language/Structured Query Language* (PL/SQL). The user interacts with the Banner database through these forms and the underlying PL/SQL. In the Banner world, users commonly refer to these forms as *Banner forms*. See Figure 3 (next page) for a high-level schematic and Figure 4 (next page) for a specific example of a Banner form.

3.2.2. The Banner Security Scheme

Like Oracle security, Banner security consists of layers. The DBA grants users access to Banner forms, and through the forms, the users have access to Banner's Oracle database.

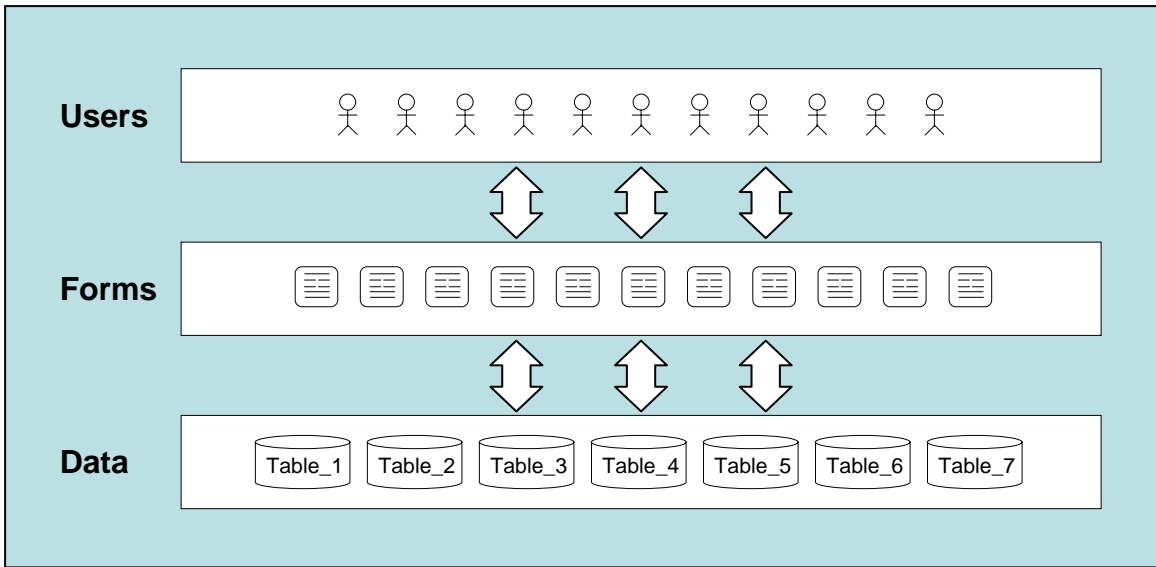


Figure 3: Users, Forms, and Database Tables

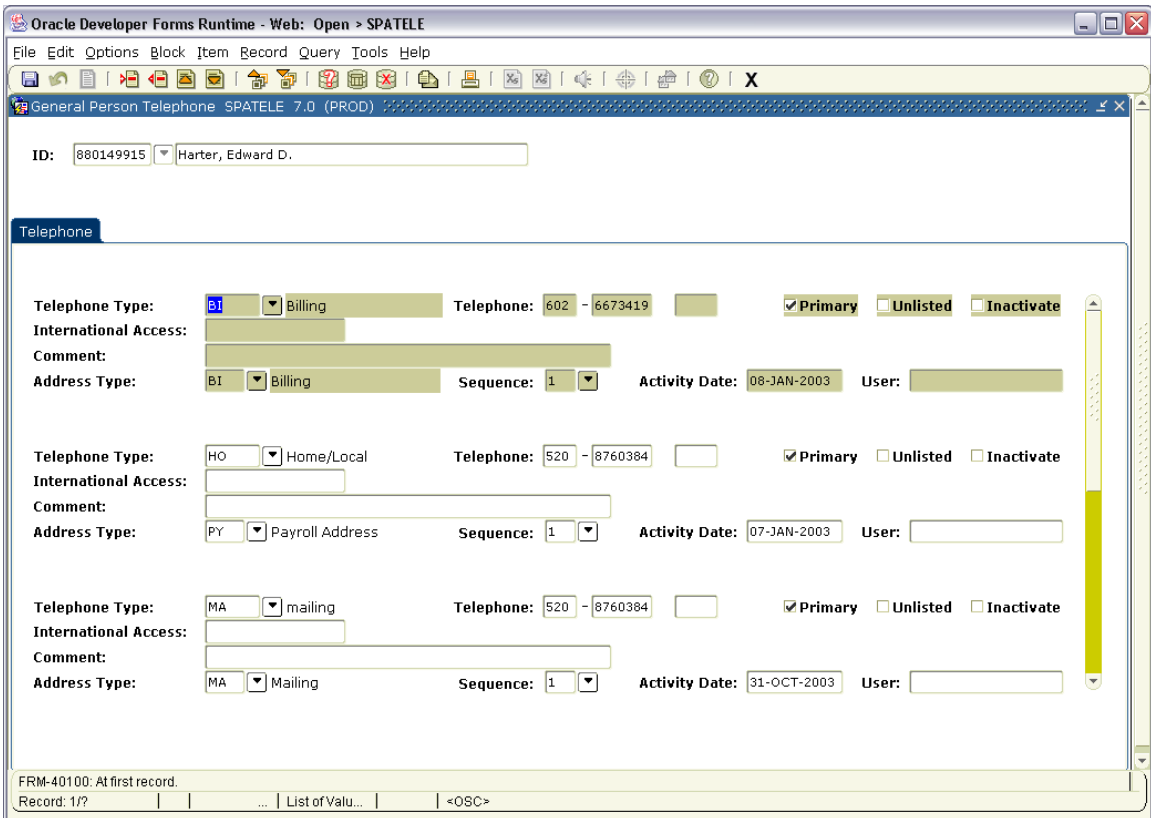


Figure 4: Example of a Banner Form

This access can be either *query access* or *maintenance access*. With query access, the user can view the data on the form. With maintenance access, the user can both view and modify the data.

User Classes. Because several users can require the same type of access to the same forms, Banner supports the concept of *user classes*. A Banner user class (UC) is analogous to an Oracle role in that it consists of user-members and access rights; but with UCs, the access rights are to forms, not to database tables directly. For example, we can have a UC named AP_CLERK. The members of the UC could be all of the clerks in the Accounts Payable department, and membership in the class could provide them with access to all of the Accounts Payable forms.

User Classes and Oracle Roles. UCs also interact with Oracle roles. For example, by virtue of membership in the AP_CLERK class, users would probably have the ability to perform select, update, delete, operations on the underlying Accounts Payable database tables. These relationships are depicted schematically in Figure 5.

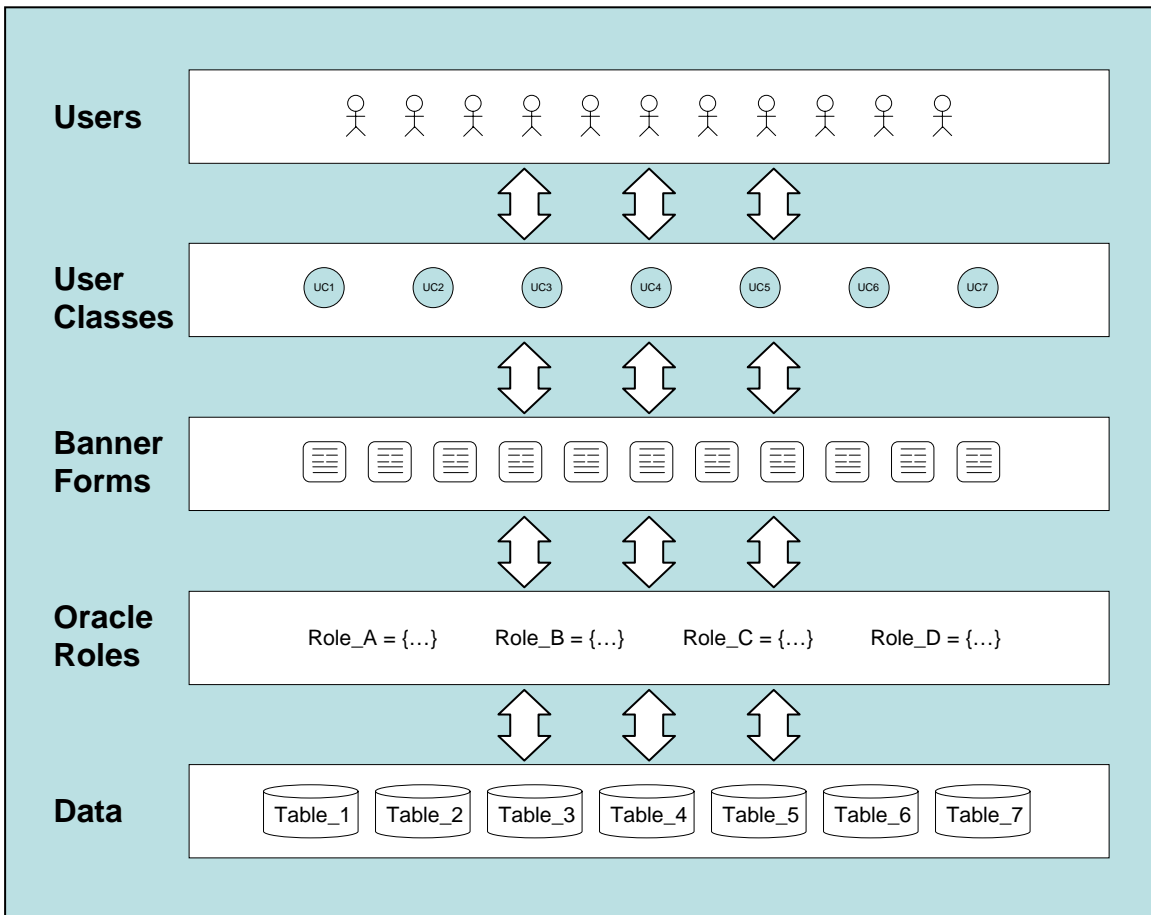


Figure 5: Banner Security Layers

As in previous figures, the double-headed arrows in Figure 5 indicate that the relationships are many-to-many: A user can belong to multiple UCs, and a UC can contain multiple users; a UC can provide access to multiple forms, and access to any given form can be provided by multiple UCs; and so on.

In conjunction with Figure 5, it is important to note that Banner grants Oracle roles *dynamically*, at run time. For example, if a given UC provides maintenance access to a particular form and therefore to the underlying database tables, the necessary roles are invoked while, and only while, the UC member is actively using the form. The Banner application issues the necessary grants ‘behind the scenes’. This mixing of data definition language (DDL) commands and the data manipulation language (DML) within an Oracle form is generally not regarded as a good programming practice. However, it works well with Banner because it provides an additional level of control.

This feature enables an authorized UC member to modify information in the database through the appropriate Banner form(s). But it does *not* enable the same UC member to log into Oracle directly and modify the information through low-level structured query language (SQL) commands.

There are three additional points to note about the scheme described in Figure 5. A procedural rule is associated with each of these points.

1. Several pre-packaged UCs are provided with the Banner system itself. The assignment of individual users to these UCs is controlled by the SUG and the DBA.

PROCEDURAL RULE: Individual users can be assigned to UCs only by the DBA—and only with the approval of the SUG and, in most cases, in response to a documented request. The processes governing approvals in this area are described in *Section 4.1.2. Banner Access Requests*.

2. It is also possible for a user organization, such as CAC, to create its own UCs to support its own specific needs.

PROCEDURAL RULE: Only the DBA can create custom UCs—and only with the approval of the SUG in response to a documented request. The request-and-approval process is described in *Section 4.1.2. Banner Access Requests*.

3. It is physically possible for the DBA to grant individual users access to a specific form or forms *directly*, without using the intermediary of UCs—just as it is possible for the DBA to grant specific database privileges to individual users directly. However, as with database privileges, the granting of access to specific forms directly to individual users can lead to clerical problems and ultimately to manageability and control problems. As a consequence, this practice is discouraged.

PROCEDURAL RULE: The DBA can grant user access to specific forms to individual users directly, but only in emergency situations—and only with an appropriate level of management concurrence and control, as described in *Section 7. Emergency Situations*.

3.3. OVERVIEW OF CASA SECURITY

Central Arizona Software Applications (CASA) is an in-house product developed by the programming staff within IRS. CASA provides a self-service platform through which

users from various departments can access custom reports and (in some instances) custom program functionality.

The CASA Main Menu is shown in Figure 6 (next page). When a user click on one of the option headings—General Users, Administration, etc.—he is taken to the appropriate sub-menu. For example, Figure 7 (next page) shows the sub-menu for *General Users*.

3.3.1. CASA and Banner Data

Like Banner, CASA relies on Oracle Forms. It also uses a companion toolset known as Oracle Reports. Most CASA programs do nothing more than provide users with the reports they need, on demand. The architecture for the CASA reporting scenario is depicted in Figure 8 (page 10). As the figure indicates, CASA report programs typically read data from Banner database tables; and in some cases they also read data from non-Banner tables—CASA tables. CASA tables reside in special CASA schemas. They do not reside in Banner schemas.

Some CASA programs also perform processing functionality. The check-in and tracking program for the Cooperative Learning Center is an example. This program checks students into the Center by reading their barcoded ID numbers from student ID cards. After it identifies a student, the program writes a record of the student's visit into the database. The program checks students out in a similar way. Thus, it documents all visits and the length of stay for each visit. It also leverages data in the Banner database. Because the student ID points directly to student records in the Banner tables, the records in CASA link each visit with the student's age, gender, academic history, and so on. This information is subsequently used in reports. This type of scenario is depicted in Figure 9 (page 10).

It is important to emphasize that CASA programs of this kind do not duplicate Banner functionality. For example, Banner knows nothing about the Cooperative Learning Center, how it does business, or what reports it needs in order to make informed decisions.

It is also important to emphasize that CASA programs of this kind do not perform update, insert, or delete operations on Banner database tables. The database writing operations depicted in Scenario 2 are restricted to non-Banner database tables that reside in non-Banner schemas.

3.3.2. CASA Data Security

CASA data security is based on Oracle data security, as it was explained in Section 3.1 above. In order to use CASA, a user needs to log into the Oracle database. As with Banner, the user logs into the database through an Oracle Forms application. In order to use a program within CASA, the user needs to have an Oracle role that provides access to the underlying database tables with which the program interacts.

In CASA, the relationship between roles and programs is typically one to many: In most cases, a single role is sufficient to run a program; but a single role can also be sufficient to run multiple programs. For example, there are several FTSE reporting programs that break the numbers out in various ways—by campus, by division, by instructional method, and so on. A single role, AR_FTSE_ROLE, provides access to all of these programs.



Figure 6: CASA Main Menu



Figure 7: CASA General Users Menu

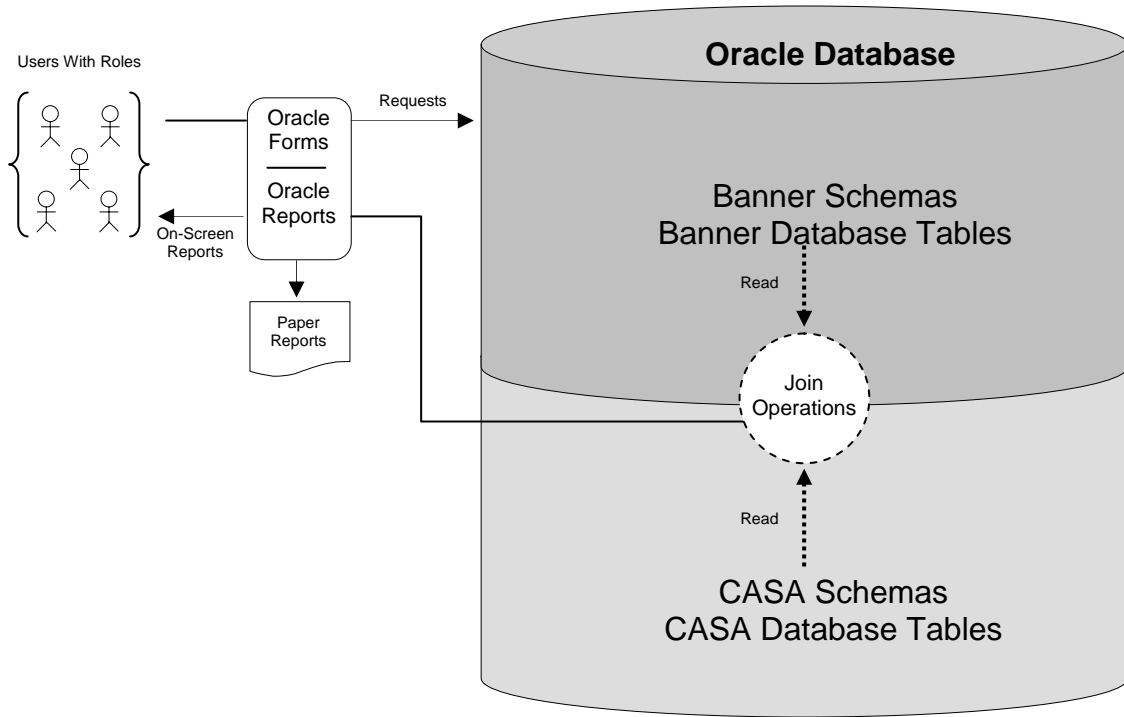


Figure 8: CASA Scenario 1—Reporting, Read Only

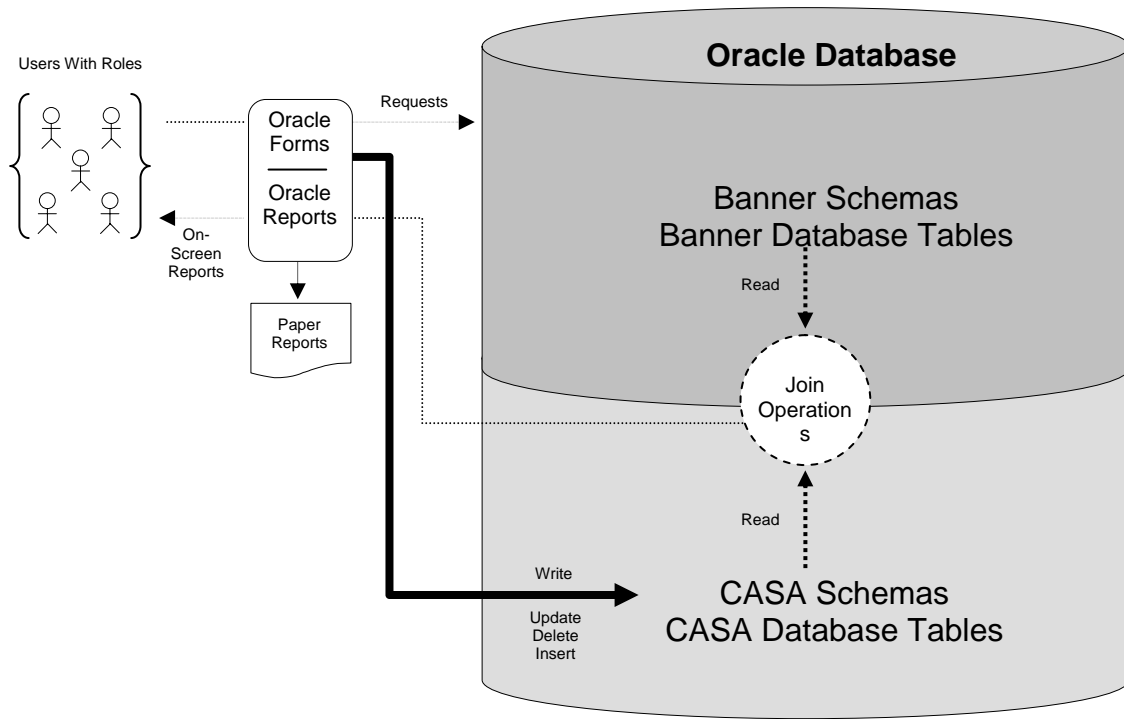


Figure 9: CASA Scenario 2—Read From Banner, Write To CASA

PROCEDURAL RULE: With the help of the Oracle Data Dictionary, the DBA maintains a cross-reference of CASA programs, Oracle roles, and CASA users.

To summarize:

- CASA programs are associated with roles.
- Roles provide read access to Banner tables for authorized users and write access to CASA tables for authorized users.
- CASA programs never write data to or delete data from Banner database tables.
- Roles are granted to users on an as-needed basis with the approval of the SUG.

Requests for CASA access are processed as described in Section 4.1.3 below.

3.4. OVERVIEW OF NETWORK SECURITY

The CAC network consists of (i) local-area networks (LANs) at individual campuses and sites and (ii) a wide-area network (WAN) through which the LANs are connected. The entire network is complex. It is made up of more than two hundred routers, switches, and other network and data communications devices. The elements most relevant to the issue of security are depicted in Figure 10.

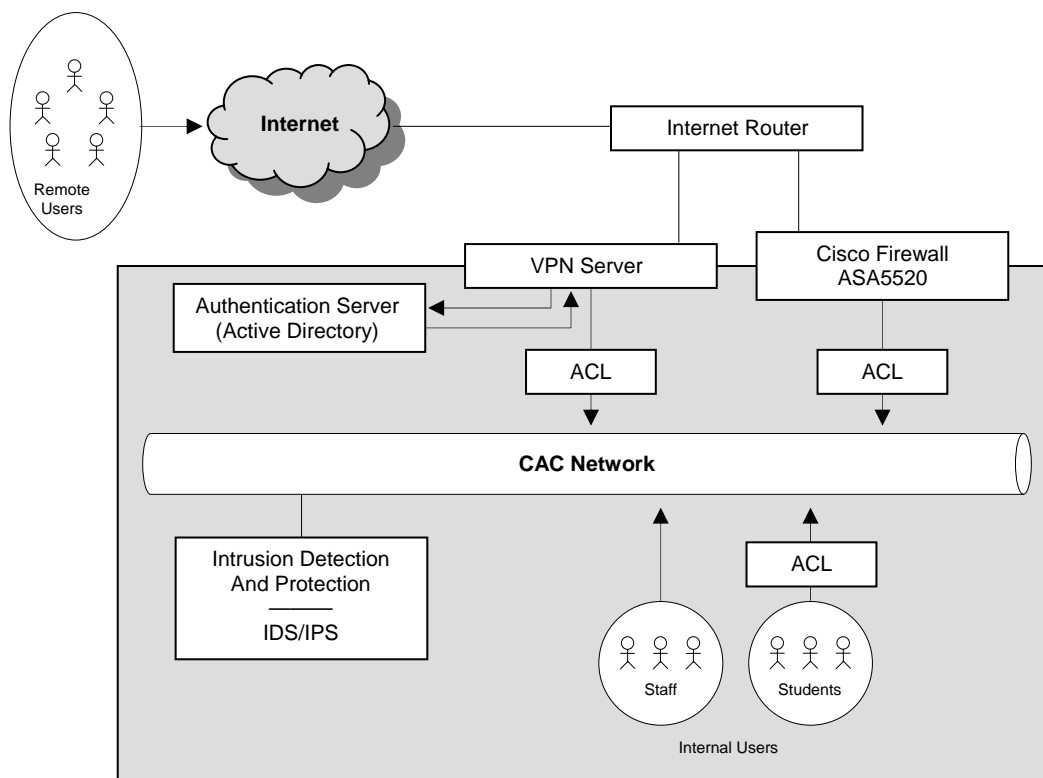


Figure 10: CAC Network Security Layout

As the figure shows, like most private networks the CAC network can be accessed both from the 'outside' and from the 'inside'. Outside or *remote* access refers to access from

locations other than CAC premises—through devices that are not directly connected or ‘hard-wired’ to the network. Conversely, *internal* access refers to access through devices that are directly connected to the network.

The gray rectangle in the figure marks the boundary that separates remote and internal access. Note that the virtual private network (VPN) server and the Cisco firewall are situated on the boundary. These elements talk directly with both the ‘outside’ and the ‘inside’.

3.4.1. Remote Access

There are actually three layers of security for remote access to the CAC network.

Layer 1: Internet Router. All Internet traffic comes through the internet router. Again, refer to Figure 10. This device routes normal Internet traffic to the Cisco firewall. It routes VPN traffic to the VPN server (see below). In addition, the Internet router detects and rejects traffic from ‘spoofers’. *Spoof* is Internet traffic with a masked or falsified Internet protocol (IP) address. The spoofer is hiding his identity. Spoofers are generally attempting to make mischief and cause harm.

Layer 2: VPN Server and Cisco Firewall. VPN is a technology that enables authorized users to ‘tunnel’ into a private network through the Internet. Once connected, they are able to work much as though they were connected to the network internally. In order to use VPN, as person needs two things: (i) VPN client software on the computer she is using, and (ii) a valid account on the VPN server which is directly connected to the CAC network. See Figure 11 for a high level schematic.

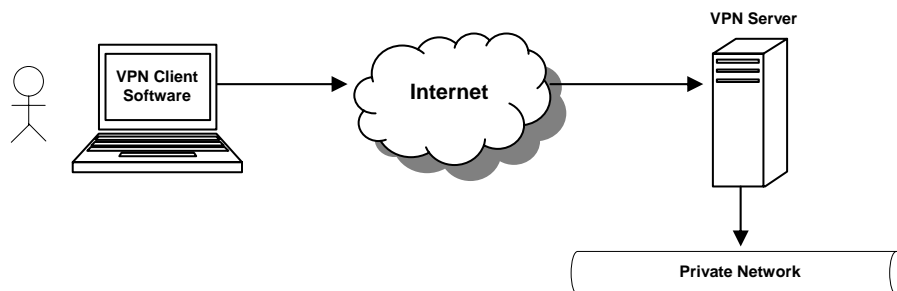


Figure 11: Remote Access Via VPN

If a person lacks either the client software or the server-side account, she cannot access the network through the VPN. VPN is a proven safe technology. The process through which CAC employees request and obtain VPN access is described in *Section 4.1.1. Network Access*.

The other element at layer 2 is the Cisco firewall. As mentioned above, all non-VPN traffic is routed through the Cisco firewall. This is an industry standard security device with a proven record of success.

Layer 3. The third layer of security is comprised of access control lists (ACLs). An ACL is a file that tells the operating system which users have access rights to which system objects—servers, file directories, individual files, and so on. Each system object has a

security attribute that identifies its access control list. In turn, the list has an entry for each user or user *group*.

3.4.2. Internal Access

Virtually all computers located on College premises are physically connected to the CAC network. Throughout this discussion, *internal access* refers to access to the CAC network from a device that is (i) located on College premises and (ii) physically and directly connected to the network. An individual user gains access to the network with her network username and password.

Refer again to Figure 10. As the figure shows, internal access controls are different for staff and for students. The login protocols for staff and students are set to connect these two groups to two different *virtual* LANs (VLANs). In other words, although students and staff use the same physical transport, they have different profiles that give them access to different sets of network-attached devices. For example, student email and staff email are hosted on different servers, and students do not have access to the district file servers used by employees. Students do not even see the Banner server on their VLAN.

3.4.3. Intrusion Detection System

CAC uses a product called *StealthWatch* for intrusion detection. This product gathers and analyzes information from all areas of the network to identify possible security breaches. These breaches include both intrusions in the strict sense (attacks from outside the organization) and misuse (attacks from within the organization). Intrusion performs *vulnerability assessment* (sometimes referred to as *scanning*). Specific functions include the following:

- Monitoring and analyzing both user and system activities
- Analyzing system configurations and vulnerabilities
- Assessing system and file integrity
- Ability to recognize patterns typical of attacks
- Analysis of abnormal activity patterns
- Tracking user policy violations

Intrusion detection follows a two-part process. The first part is host-based and is considered the *passive* component. This includes inspection of the system's configuration files to detect inadvisable settings; inspection of the password files to detect inadvisable passwords; and inspection of other system areas to detect policy violations. The second part is network-based and is considered the *active* component. In this step mechanisms are set in place to address known methods of attack and to record system responses.

The system is capable of producing reports and issuing alerts in response to suspect activity on the network. In some instances, it is capable of responding and taking remedial action directly.

4. ACCESS REQUESTS AND APPROVALS

All access requests must be made through *Central Arizona Request Services (CARS)*, which is a module within *CASA*. *CARS* is a set of in-house Oracle database applications through which users make requests for services of various kinds. Requests for network access are made through the *Network Access Request Function (NARF)*; requests for data access are made through the *Data Access Request Function (DARF)*. See Figure 12.

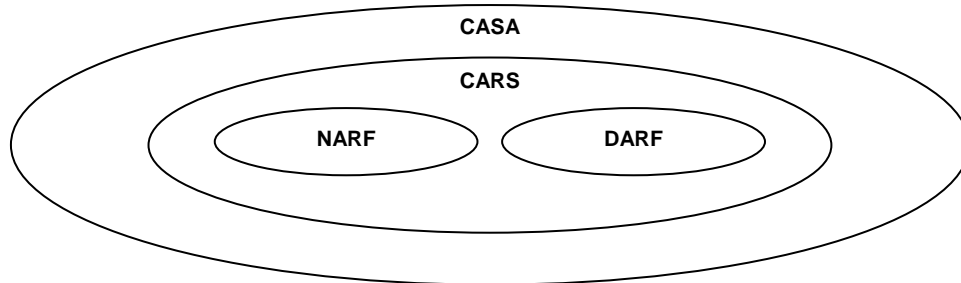


Figure 12: CASA, CARS, NARF, and DARF

Like all *CASA* programs, *CARS* leverages the data from the Banner database (but it never modifies data in the Banner database). When a user logs into any *CARS* module, the Oracle DBMS knows who that user is. Specifically, it knows that the user is a fulltime employee of the College; it knows the person's employee ID; her office telephone number; her office address; and so on.

CARS keeps database records of all requests made—who made them, their disposition (approved or disapproved), to whom the work was assigned, and when the work was completed. *CARS* request records are maintained in *CASA* database tables, not Banner tables. *CARS* programs operate on the model of 'CASA Scenario 2', as depicted in Figure 9 above (page 10).

For more information about *CARS* and how it is used in this context, see the *Network Access Request Function (NARF)* and *Data Access Request Function (DARF)* Sections of the *CARS User Manual*.

4.1. NORMAL REQUEST PROCESSING

Rules and processes are in place for granting (i) network access, (ii) Banner access, and (iii) *CASA* access. The details of these processes are slightly different from one another and will be explained separately. Note that network access is required for both Banner and *CASA* access. Accordingly, network access will be described first.

4.1.1. Internal Network Access

Fulltime Employees. Fulltime employees are granted access to the CAC network by default. There is no request-and-approval process for granting network access to fulltime employees, and every fulltime employee is given a username and a password which are activated on the employee's first day of work. The standard format for usernames is

Firstname.Lastname

For example

john.smith

Passwords expire after ninety days. The user is notified when his password is about to expire and is directed to change it.

NOTE: Network access is necessary for Banner and CASA access, but it is not sufficient. Everyone with network access can access his email and the shared file servers, but not necessarily either Banner or CASA. See Subsections 4.1.2 and 4.1.3 below.

Part-Time Employees. Part-time employees are granted network access (and email) if they need it. Requests for network access for part-time employees are to be submitted through NARF/CARS by the employee's manager or department head.

The request and approval process is as follows:

- Step 1: The appropriate manager or department head logs the request into CARS. The name of the requestor, the date and time of the request, and the request itself are recorded in the database. A tracking number is assigned to the request, and an email is generated to the requestor to tell her that the request has been logged and to notify her of the tracking number.
- Step 2: After the request is logged into CARS, the members of SUG are notified by email.
- Step 3: Each SUG member pulls up the request online, in CARS.
- Step 4: If a SUG member approves, she checks the *Approve* box.

If a SUG member believes the request requires discussion, she calls a SUG meeting. (This meeting can take place by teleconference with each member viewing the request online in CARS.)

If all members approve the request, they all check the *Approve* box in CARS. A database record is written to document the date and time of approval. An email is generated to the requestor to notify her that the request has been approved, and an email is generated to the Network Administrator. The Network Administrator then pulls up the request in CARS and implements the request. When he is finished, he marks the request *Completed* in CARS. A database record is written to document the date and time of completion, and an email is generated to the requestor to notify her that the work has been completed.

If the SUG decides not to approve the request, all members check the *Disapprove* box in CARS. As they save the record in the database, a dialog box appears in which they enter the explanation for disapproval. When the record is committed to the database, an email is generated to the requestor. The text of the email contains the explanation for disapproval.

PROCUDURAL RULE: The criterion for approval of network access requests for part-time employees is that such access must be essential for job productivity. Personal convenience is not a sufficient reason.

4.1.2. Remote Network Access

VPN access is not automatically granted to anyone. The request-and-approve process for VPN access for all employees is the same as the four-step process described above for network access for part-time employees. The request for VPN access must be initiated by the employee's manager or department head.

PROCUDURAL RULE: The criterion for approval of VPN access requests is that such access must be essential for job productivity. Personal convenience is not a sufficient reason.

4.1.2. Banner Access Requests

Normal Banner access requests can be broken out into five types. It is useful to understand the differences between these types, although the process for dealing with four of them is the same. The five types are:

1. Requests for access for new users.
2. Requests to modify the Job/UC cross-reference table in CASA. (See below for a description of what this table is and what it does.).
3. Requests to add or remove an existing user to or from a UC.
4. Requests to modify the access provided by a UC—e.g., to grant access to an additional form, or to revoke access to an existing form.
5. Requests to create a new UC and add users to it.

As regards Type 1, most new users are new employees to the College. However, in this context 'new user' actually refers to *employee new to a job*. For example, if a person was formerly an accounts receivable clerk and has been reassigned as a payroll clerk, he is considered to be a *new user* in this sense.

Banner Access Requests for New Users. Access-request processing for new users is largely automated. Under the direction of SUG, HR maintains a cross reference of job IDs and UCs in CASA. Note that the relationship between jobs and UCs is (at least potentially) many-to-many because people with different jobs can require access to overlapping sets of Banner forms. See Figure 13 (next page) for a conceptual illustration.

PROCEDURAL RULE: The Job/UC cross reference is hosted in database tables in CASA. HR maintains the cross reference through an Oracle Forms interface in CASA. Changes to the cross reference tables can be made only by HR and only with SUG approval of a Type 2 change request. NOTE: If the 'new user' is an existing employee with a former job, the DBA is also instructed to revoke all of his 'old user' access.

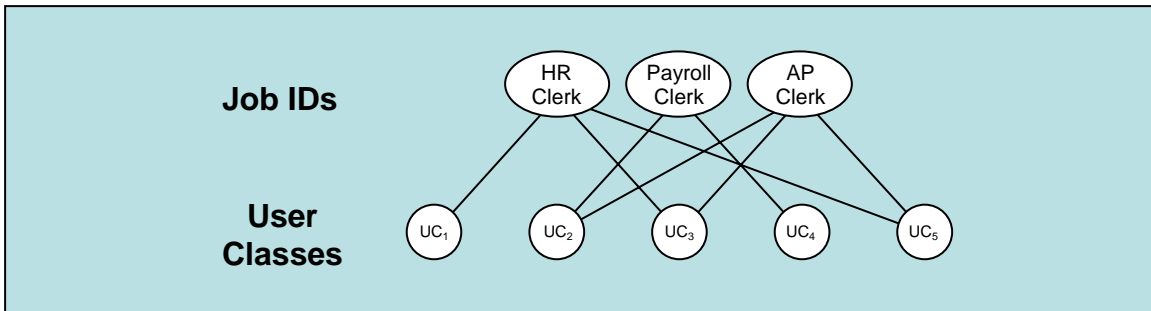


Figure 13: Many-to-Many Relationship Between Jobs and User Classes

When HR assigns the new user to her job, a Banner access request is automatically generated. The specific contents of the request are culled from the Job/UC cross-reference table in CASA. Because the SUG is the ultimate approver of the cross-reference table, the request is, by implication, automatically approved. Accordingly, the request is automatically logged into the CARS database with an *Approved* status, and an email is automatically sent to instruct the DBA to implement the request. See Figure 14

<u>Step Zero</u>	<u>Step 1</u>	<u>Step 2</u>	<u>Step 3</u>
Job/UC cross-reference exists	New User assigned to job Cross-reference links user to UCs	Access request auto-generated and auto-approved DBA notified	DBA implements request

for a delineation of the process steps.

Figure 14: Banner Access for New Users

Other Banner Access Requests. ‘Other Banner Access Requests’ refers to requests of Types 2, 3, 4, and 5. As regards Type 3, note that requests to remove a user from a UC can be made for either of two reasons: either (i) because the user no longer needs the access in order to perform his job; or (ii) because the user has been terminated as an employee of the College. In both cases, it is the responsibility of the manager or of HR to initiate the request.

The normal approval process for Banner access requests of Types 2, 3, 4, and 5 is as follows:

- Step 1: The appropriate manager or department head logs the request into the DARF function within CARS. The name of the requestor, the date and time of the request, and the request itself are recorded in the database. A tracking number is assigned to the request, and an email is generated to the requestor to tell her that the request has been logged and to notify her of the tracking number.

NOTE: It is the requestor's responsibility to know what she is requesting. In other words, it is her responsibility to know what the UCs are and what levels of access they provide. An up-to-date list of UCs and the levels of the access they provide can be obtained from CASA.

Step 2: After the request is logged into CARS, the members of SUG are notified by email.

Step 3: Each SUG member pulls up the request online, in CARS.

Step 4: If a SUG member approves, she checks the *Approve* box.

If a SUG member believes the request requires discussion, she calls a SUG meeting. (This meeting can take place by teleconference with each member viewing the request in CARS.)

If all members approve the request, they all check the *Approve* box in CARS. A database record is written to document the date and time of approval. An email is generated to the requestor to notify her that the request has been approved, and an email is generated to the DBA. The DBA then pulls up the request in CARS and implements the request. When he is finished, he marks the request *Completed* in CARS. A database record is written to document the date and time of completion, and an email is generated to the requestor to notify her that the work has been completed.

If the SUG decides not to approve the request, all members check the *Disapprove* box in CARS. As they save the record in the database, a dialog box appears in which they enter the explanation for disapproval. When the record is committed to the database, an email is generated to the requestor. The text of the email contains the explanation for disapproval.

The process flows for Banner access requests and approvals are depicted in Figure 15 (next page). Note the difference between Type 1 and the others.

4.1.3. CASA Access Requests

The processing of CASA access requests is similar to the processing of Banner access requests, but some details are different because CASA access is based on Oracle roles rather than Banner UCs. Also, CASA access requests are stated in terms of *programs*—for example, ‘User A needs to be able to run Report B’—not in terms of the underlying database privileges or of roles. Requestors are not required to frame their requests in terms of Oracle roles because they typically do not know what the roles are, or even that such things as Oracle roles exist. As stated in Section 3.3.2 above, the DBA maintains a cross-reference of CASA programs and their required roles. Accordingly, if a manager requests for User A to access Report X, the DBA will know which Oracle roles are required.

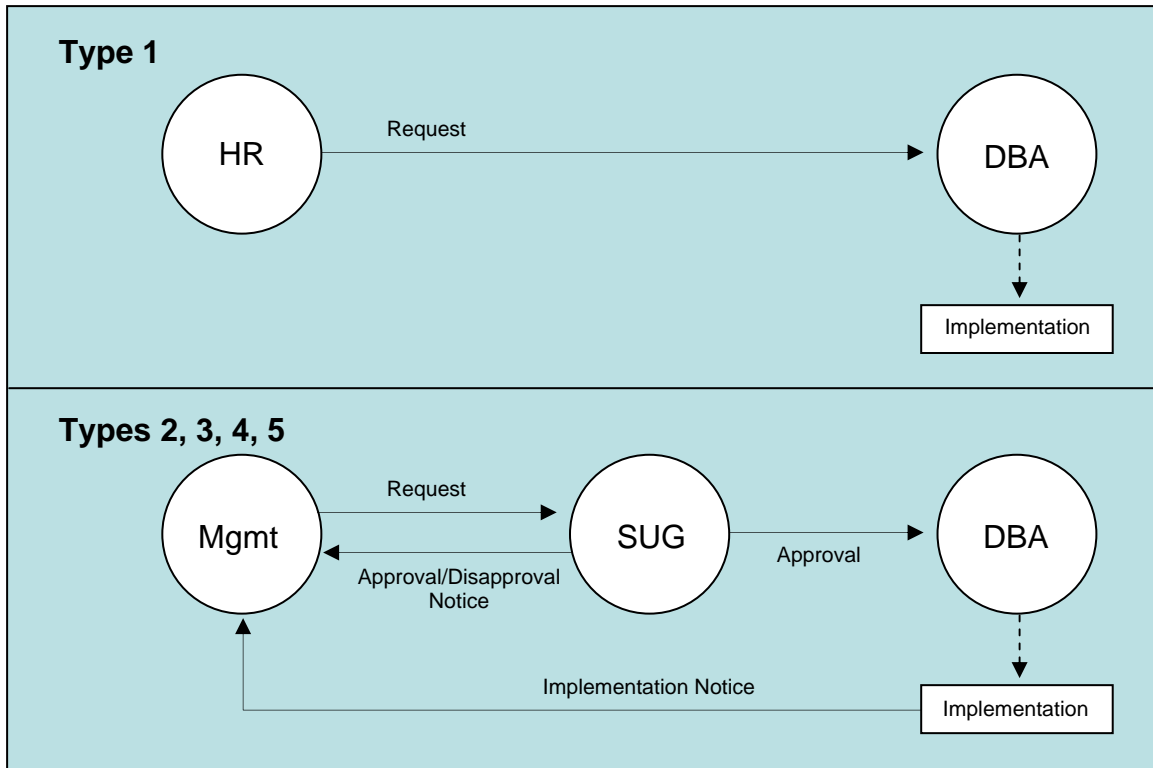


Figure 15: DARF Request Processing

The basic process for dealing with normal CASA access requests is as follows:

- Step 1: The appropriate manager or department head logs the request into DARF/CARS. The name of the requestor, the date and time of the request, and the request itself are recorded in the database. A tracking number is assigned to the request, and an email is generated to the requestor to tell her that the request has been logged and to notify her of the tracking number.
- Step 2: After the request is logged into CARS, the members of SUG are notified by email.
- Step 3: Each SUG member pulls up the request online, in CARS.
- Step 4: If a SUG member approves, she checks the *Approve* box.

If a SUG member believes the request requires discussion, she calls a SUG meeting. (This meeting can take place by teleconference with each member viewing the request in CARS.)

If all members approve the request, they all check the *Approve* box in CARS. A database record is written to document the date and time of approval. An email is generated to the requestor to notify her that the request has been approved, and an email is generated to the DBA. The DBA then pulls up the request in CARS and implements the request. When he is finished, he marks the request *Completed* in CARS. A database record is written to document the date and time of completion, and an email is generated to the requestor to notify her that the work has been completed.

If the SUG decides not to approve the request, all members check the *Disapprove* box in CARS. As they save the record in the database, a dialog box appears in which they enter the explanation for disapproval. When the record is committed to the database, an email is generated to the requestor. The text of the email contains the explanation for disapproval.

Another difference between the processing of CASA access requests and Banner access requests is this: If a user requests the creation of a new program in CASA—a new report, for example—he is (by implication) requesting the ability run the program. Accordingly, if the request is approved, the DBA creates the necessary role (if necessary) and grants it to the requestor.

Requests for new CASA programs are made in CARS, through the *Programming Services Request Function (PSRF)*. (*PSRF* is pronounced like the word ‘surf’.) For more information on the PSRF module in CARS, see the *Programming Services Request Function (PSRF)* section of the *CARS User Manual*. Additional information is also provided in this document, in *Section 6.2. Changes to CASA*.

4.2. SPECIAL REQUEST PROCESSING

In emergency situations the normal rules and processes for dealing with access requests are often impracticable. For a description of how access requests are handled in these cases see *Section 7. Emergency Situations*.

5. ONGOING MONITORING AND MAINTENANCE

The following pieces of information are reviewed by the SUG on a bi-monthly basis to ensure that access to Banner data is available to those who need access it, but not to others.

- The Job/UC cross reference.
- The dataset consisting of
 - The active UCs
 - The Banner forms to which the UCs provide access
 - The users who belong to the UCs
- The dataset consisting of
 - The active Oracle roles
 - The database tables to which the roles provide access
 - The users who have the roles granted to them.

Omissions and erroneous inclusions are noted and corrected.

6. SOFTWARE CHANGES

Software changes include changes to Banner programs and changes to CASA programs. Changes to CASA programs include the introduction of new CASA programs and the modification of existing ones.

6.1. CHANGES TO BANNER

Changes to Banner programs are made almost exclusively through the application of patches and upgrades provided by the vendor. The general rule at CAC is not to make in-house programming modifications to Banner programs. This is done only in rare circumstances.

6.1.1. In-House Modifications

Modifications to Existing Banner Programs. An example of an in-house modification is the work that was done on Banner's paycheck printing program. As provided by Banner, this program prints both the employee's ID and his social security number on the stub of his paycheck. The printing of the social security number was regarded as highly objectionable by several employees, by HR, and by the College President. The vendor was contacted and they indicated that they had no plans to change their paycheck printing program to comply with these concerns. Accordingly, the programming staff modified the program to suppress the printing of the employee's social security number.

Database Triggers on Banner Tables. Another type of in-house modification to Banner is the creation of database 'triggers' on Banner tables. A trigger is a program that runs (or 'fires') automatically in response to a database event. For example, an *update* trigger on Table A is a program that runs each time a database update is performed on Table A.

Strictly speaking, in-house triggers are not really modifications to existing Banner programs or to existing Banner functionality. In-house triggers are separate programs that provide functionality that the Banner programs do not address. By the same token, in house triggers do change the way in which Banner functions overall.

For example, in 2006 the administration decided that the College should provide students with email. It was also agreed that email *addresses* would be given to all matriculated students, but that live email *accounts* would be given only to students who actually registered for classes. The most straightforward way of accomplishing this without creating an overwhelming amount of manual work was to create two database triggers.

The first is a trigger on the Banner student table (SGBSTDN). When a student is admitted into the College, this trigger creates an email *address* (not an email account) for the student and writes a record for the student in the Banner email address table (GOREMAL). Student email addresses are of the form

Firstname_Lastname@stu.centralaz.edu

If the Firstname_Lastname combination creates a collision with another student—for example, if there is more than one John Smith—the trigger adds numeric digits to the Firstname_Lastname string to make it unique. Keeping with the John Smith example, we might have a John_Smith@stu.centralaz.edu, a John_Smith1@stu.centralaz.edu, a John_Smith2@stu.centralaz.edu, and so on.

The second is a trigger on the Banner registration table (SFRSTCR). When a student registers for a class, this trigger performs a check to determine whether or not he already has an email account (not just an email address). If he does, nothing happens. If he does not, the trigger reads the student's email address from GOREMAL and writes a record for the student in a special email request table in CASA. Another program trolls this

request table every twenty minutes and creates student email accounts in Microsoft Exchange.

PROCEDURAL RULE: Database triggers on Banner tables are not created in Banner schemas. They are created in, and they reside in, special schemas set up in CASA. They ‘fire’ when the designated Banner tables are modified during normal business processing.

Documentation of Changes. All changes of the kind just described are made only in response to requests made through PSRF requests that are approved by the SUG. As mentioned in Section 4.1.2, PSRF is a module within CARS. Records of all PSRF requests, approvals, and completions are stored in the CARS database.

Moreover, all changes that alter Banner functionality (including the creation of database triggers) are documented and recorded in the *Banner In-House Change Log*.

PROCEDURAL RULE: All in-house programming that alters Banner functionality is documented and recorded in the *Banner In-House Change Log*; and all programming source code is maintained under strict configuration and change control in VSS.

6.1.2. Vendor-Provided Patches and Upgrades

As stated at the beginning of this section, the majority of changes to Banner are made through the application of vendor-supplied *patches* and vendor-supplied *upgrades*.

Patches. Patches are bug fixes. These are applied by the DBA as they are made available from the vendor. The DBA is notified of the availability patches on the vendor’s website, which the DBA routinely monitors.

Upgrades. Upgrades are of two types: (i) Some are nothing more than bundles of patches. The process for applying these upgrades is the same as described in the preceding paragraph under *Patches*. (ii) Other upgrades contain enhancements and embellishments to Banner functionality. The process for applying these upgrades is as follows:

- Step 1. The DBA learns of the availability of the upgrade by monitoring the vendors website.
- Step 2. The DBA confers with the Banner lead in the affected business organization.
- Step 3. If the affected business organization does not want the upgrade to be applied, a representative of the organization submits a BURF request not to apply the upgrade. The request is either approved or disapproved. In either case, the decision is documented in the CARS database; the requestor is notified by email; and the process stops.

Otherwise, if the affected business organization does want the upgrade, a representative of the organization submits a request to perform the upgrade through BURF, which is a module within CARS. If the request is disapproved, the decision is documented in the CARS database; the requestor is notified by email; and the process stops.

- Step 4. If the request is approved, the DBA applies the upgrade in the Banner TEST environment and negotiates a testing schedule with the lead user.
- Step 5. When testing is complete, the lead user notifies the DBA and signs off on a *Banner Upgrade Testing Form*. (At the time of release of this document, the *Banner Upgrade Testing Form* is a paper form, but plans are in place to make this step an integrated part of the online BURF process within CARS.)
- Step 6. The DBA applies the upgrade in the Banner PROD environment.

The BURF process is similar to the process for dealing with Banner data access requests, but because the request can be for non-action, the laws of combinatorics dictate that the possible number of outcomes is greater. (For documentation purposes, the inclusion of a request for non-action is important.) When the request is reviewed by the SUG, there are four possible cases:

- Case 1. The request is to forego the upgrade and the SUG approves it. In this case a record is written to the CARS database to document the approval; and emails are automatically generated to the requestor and the DBA. But no work order is generated for the DBA.
- Case 2. The request is to forego the upgrade and the SUG disapproves it. In this case a record is written to the CARS database to document the disapproval, and emails are automatically generated to the requestor and the DBA. No work order is generated for the DBA, but it is likely that this action will probably lead to a repetition of the seven-step process described above.
- Case 3. The request is to apply the upgrade and the SUG approves it. This is the most common case. In this case, a record is written into the CARS database to document the approval; emails are automatically generated to the requestor and the DBA; and a work order is generated for the DBA.
- Case 4. The request is to apply the upgrade and the SUG disapproves it. In this case, a record is written into the CARS database to document the disapproval, and emails are automatically generated to the requestor and the DBA. No work order is generated for the DBA.

It is essential that upgrade requests be reviewed by the cross-functional SUG—for two reasons: (i) changes to one module can affect users and business processes in organizations that do not use the module directly; and (ii) an upgrade to Module A might require a companion upgrade to Module B, where Module A and Module B are ‘owned’ by different functional organizations.

6.2. CHANGES TO CASA

As mentioned at the beginning of this section, changes to CASA consist of (i) requests for new CASA programs and (ii) requests for modifications to existing programs.

CASA change requests are processed through the PSRF module in CARS. The PSRF process is similar to the process for dealing with Banner data access requests: Requests are logged into CARS through PSRF. Requests are reviewed by the SUG. If approved, requests generate work orders that are assigned to a member of the technical staff; and an

email is automatically generated to the requestor to notify him that his request has been approved and to whom the work has been assigned. If disapproved, an email is automatically generated to the requestor to notify him of the disapproval and the reason for it.

When the assignee completes the work, she marks it *Completed*. The completion is logged in the database and an email is automatically generated to the requestor to notify him of the change in status. See Figure 16 (next page) for a graphical representation of the process flow.

This process differs from the DARF process for access requests in that the SUG holds a weekly face-to-face meeting to review and approve PSRF requests. As with other processes within CARS, it is key to note that every action in this process is documented and timestamped in the CARS database tables.

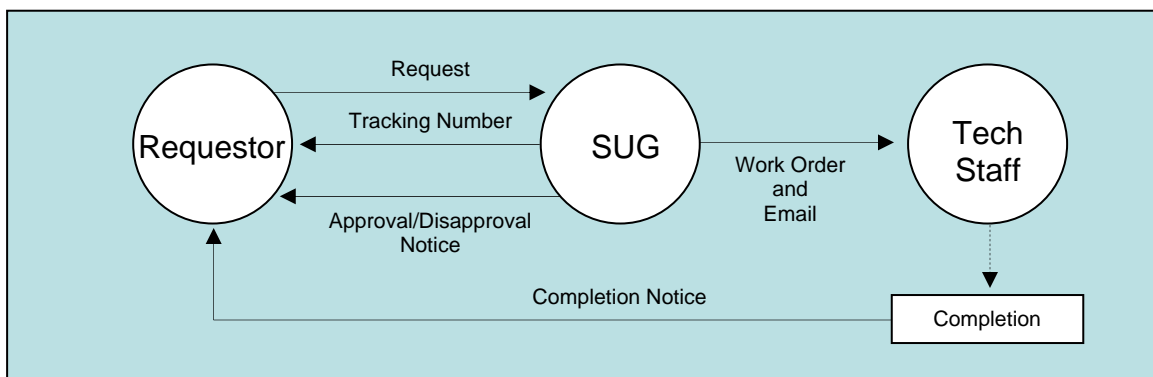


Figure 16: PSRF Request Processing

PROCEDURAL RULE: All programming that alters CASA is approved by the SUG, and all programming source code is maintained under strict configuration and change control in VSS.

7. EMERGENCY SITUATIONS

An emergency situation is one which can seriously impede or compromise business operations if not addressed and resolved within one business day. Emergency situations are typically dealt with by the DBA or a programmer, or the DBA and a programmer working together.

Emergency situations, and the need to respond to them, can arise in conjunction with any of the activities described in this document—activities dealing with Banner access, CASA access, programming, vendor upgrades, and so on.

Effective response to emergency situations requires quick action, but it also requires an appropriate level of management approval and oversight. The process for dealing with these situations at CAC is as follows:

- A user notifies the IRS Help Desk of the situation.
- Technical staff determines the technical options for addressing the situation.

- Approval is sought from the Cabinet-level manager who presides over the affected business area.
- If that Cabinet-level manager is not available, approval is sought from another Cabinet-level manager.
- If no Cabinet-level manager is available, approval is sought from the administrator-level manager who presides over the affected business area.
- If that administrator-level manager is not available, approval is sought from another administrator-level manager.
- Technical staff discusses the required action(s) with the approver.
- When agreement is reached, technical staff performs the work.
- For documentation purposes, a request is entered into CARS in the name of the approving manager and is forced through the system with a priority of *Emergency*.