System Administration
Training Guide

S110 Security Management

Acumatica

Table of contents

User Security

Learning Objects:

- *Learn different ways to secure Acumatica ERP*
  - *User authentication*
  - *User authorization*
  - *Password policies*
  - *Roles and memberships*

The User Security module provides functionality you can use to administer user access to information and set up various security policies.

The User Security module in Acumatica ERP provides functionality you can use to administer user access to information and set up various security policies. Also, you can update and customize documentation, manage announcements and various file attachments, launch or stop background processing of system tasks, and much more.

User Authentication and Authorization

Acumatica administers access based on roles, which ensures security and separates user duties. Users belong to custom-configured, task-oriented roles, and you can manage each role's access to specific forms, fields, and actions. Also, Acumatica lets you restrict users' access to particular vendor and customer accounts, General Ledger accounts, subaccounts, subaccount segment values, budget articles, and other objects. You can create any number of restriction groups that include users and some of the most sensitive objects. Users assigned to one restriction group cannot view the objects assigned to another restriction group.

User Authentication

Acumatica ERP requires users to authenticate by using the appropriate username and password. After successful authentication, user membership in roles and restriction groups is checked. Then users may access only the resources and perform only the actions they are authorized to by their roles and restriction groups. A user with no role assigned has no access to the system.

If Acumatica ERP in your organization is integrated with Active Directory, so that the users, who have accounts in the local network domain, can log into Acumatica ERP using their domain credentials, user accounts are managed from Active Directory and are not affected by user account and password policies set in Acumatica ERP.

User Account and Password

In Acumatica ERP, users are defined as internal users (those who are part of your organization) and external users (those who are not part of your organization, but who might be given some access to information). Each account for either type of user should include a username, a password, and other required properties, such as the user's first and last name, email address, password policy options, and membership in roles.

Additionally, company policy may require that remote users be allowed to access the system only from a particular range of IP addresses. If such a user attempts to access the system from a computer with an IP address that outside of the specified range, access is denied.

The system administrator creates an account for an internal and external users on the Users (SM.20.10.10) form.

Any user password (whether it is temporarily set by a system administrator or set for longer-term use by the user) must comply with the organization-wide password controls and the password policy options set for the individual user.

Organization-Wide Password Policy Options

The Security Preferences (SM.20.10.60) form lets system administrators and other authorized users set controls, including those related to passwords, for all users. These requirements affect all user accounts, both internal and external.

Options on the form allow implementation of the following organization-level policy decisions:

- Whether to force users to change passwords and, if so, how often. For maximum security, we recommend that users change passwords periodically, such as every 90 to 180 days. Shorter ranges can reduce the security of accounts because users may struggle to create complex, memorable passwords often, which encourages them to write down passwords or use simpler ones.
- Whether to store passwords encrypted in the database and, if so, what type of encryption to apply.
- Whether passwords must meet complexity requirements. If complexity is required, passwords must have at least three of the following features: lowercase letters, uppercase letters, special symbols, and digits.
- Whether the minimal length requirement is applied and, if so, what particular length is required.
- Whether to apply an additional password validation mask and, if so, which mask.

Acumatica ERP provides account lockout functionality to prevent unauthorized access to the system. After a user has had a specified number of unsuccessful attempts to log in, the user's accounts can be temporarily locked out for the duration you have specified on the Security Preferences form.

Also, you can specify a time interval after which the lockout counter is reset. If a user attempts to log in but does not exceed the maximum number of attempts, after this time interval, the counter of unsuccessful attempts will be reset to zero.

Password Policy Options for Individual Users

To set up password policy for a particular user, use the Users (SM.20.10.10) form. You can use the same password policy for all users or create different policies for individual users. Options on this form control whether the user:

- Must change the password on the next login after the account is set up. This option ensures that only the user has access to the account.
- May change his or her password at will.
- May have a password that never expires. You should not use this option for all user accounts because it will result in reduced security for the site. It might be appropriate for some users, however.
- May recover his or her password.

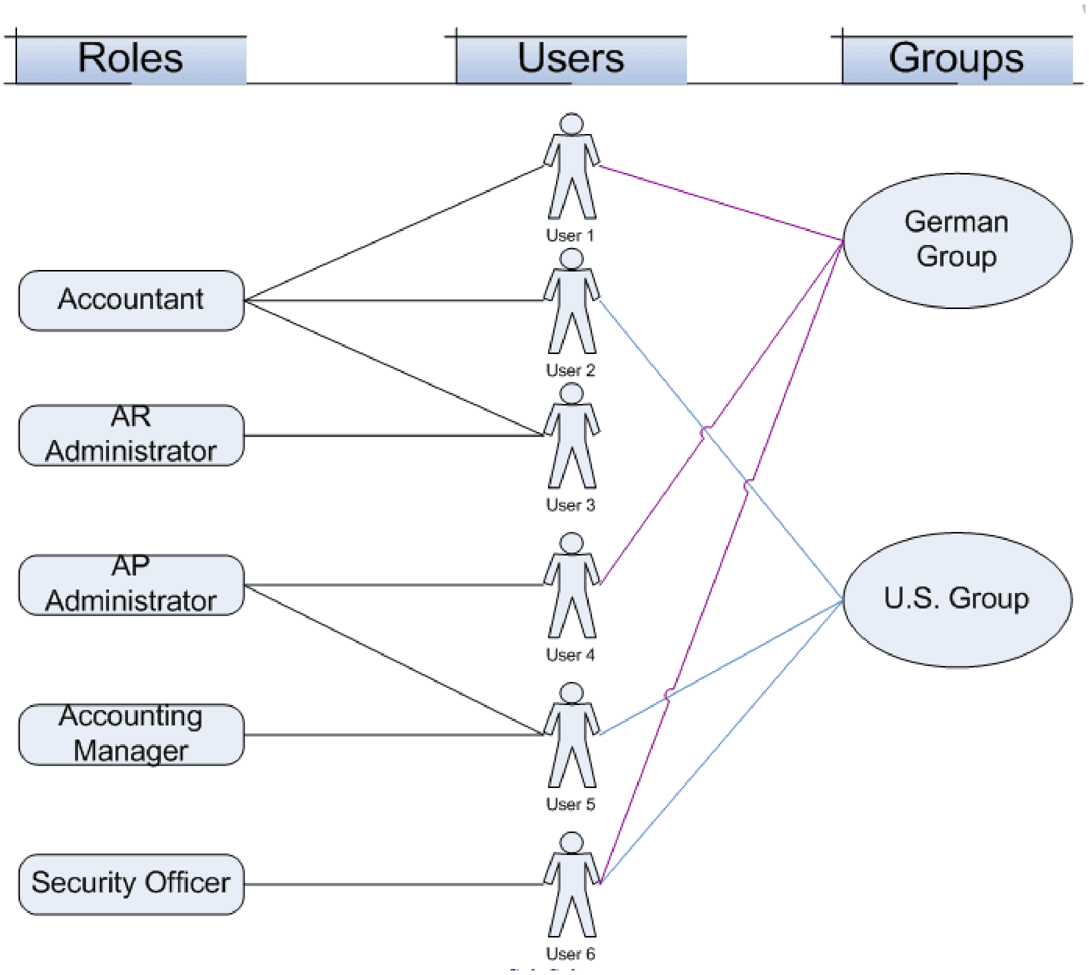
If the organization's password policy allows password recovery for the user, the user should also specify a password recovery question and an answer to this question. If the user forgets the password, he or she will be prompted to answer the password recovery question. If the user gives the correct answer, the system will generate a new password and send it to the user's email address.

Users can change their passwords or edit their information using the User Profile… (SM.20.30.10) form.

User Authorization

While the authentication procedure determines which users may log into the system, the authorization procedure determines whether a user's membership in roles and restriction groups facilitates access to a particular resource. After the user provides a valid username and password, the system checks the user's membership in roles and restriction groups. The user then has access to system modules, forms, fields, and actions based on roles assigned to him or her. Access to database records created with the help of specific forms is defined by access to the named forms. Thus, if a role grants or denies access to the form, it grants or denies access to all of the records of the same type generated using this form. If the user has multiple roles that have different levels of access rights to an entity, the highest level applies.

In Acumatica ERP, restriction groups manage access to some of the records or entities of multiple types, such as General Ledger accounts, subaccounts, subaccount segment values, budgets, vendor and customer accounts, warehouses, inventory items, and email accounts. Membership in a restriction group restricts access to only specific entities included into the group, although the role may allow access to all the named entities. See the following illustration of the relationship between roles, users, and restriction groups.



Role-Based Access

Acumatica ERP uses a role-based approach to security. You assign users to one or more roles, which are then granted various access rights to system objects. You can quickly and easily control access to system objects because changing a role's access rights affects all the users assigned to that role.

To log in to the system, a user must provide a valid username and password. After a successful login, the system checks the user's membership in roles. The user may view only the forms and fields, and perform only the actions, permitted by the user's roles. A user with no role assigned has no access to the system.

Users are divided into two groups by means of **Linked Entities**, with roles assigned accordingly:

- **Employee**: An employee-related user type is associated with employees in your system. For such a user type, you select Employee in the Linked Entity box on the User Types (EP.20.25.00) form. These user types are intended for users who are internal to your company (generally employees of your company and possibly consultants that you consider part of your company).

- **Contact**: A contact-related user type is associated with a contact in your system. You indicate such a user type by selecting Contact in the Linked Entity box on the User Types (EP.20.25.00) form. These user types are intended for users who are external to the company—for example, partners or contacts. As opposed to employee-related user types, you can use contact-related user types to delegate the right to create users.

In Acumatica ERP, you can also create a user that is not associated with any user type. Generally, such users are treated the same way as those with an employee-related user type, although they don't have default roles assigned and can be assigned any roles available in the system. Also, the users that are not associated with any user type can create new users of any user type.

Users hold different positions in the company and have different responsibilities. While they work with the system, users perform different tasks, each of which requires different access rights to financial and other modules, forms, records and operations over records. Users in similar positions generally perform similar tasks. Rather than assigning access rights to each user for each object the user must access, you assign typical sets of access rights to roles and then grant these roles to users.

Roles help you easily manage access rights for all users of the system. Changing one role can alter access rights for many users. Properly configured roles are not affected when new users are hired or existing users quit, and you can easily add a user to a role or remove a user from one. Give each role only the access rights necessary to perform typical tasks. To keep the number of roles manageable, it is better to give a user multiple roles than to create complicated roles that overlap with already-defined ones. For example, an Accounting Manager role should have broader access rights than the Accountant role. Instead of giving the Account Manager the same privileges the Accountant role has, give a user in a managerial position the Accountant role along with the Accounting Manager role.

⚠ Acumatica ERP has the built-in role Administrator, which provides full access to all system objects. It is recommended that you use this role only during initial system setup to define roles and enter users. Then use the role only in extraordinary cases. Do not modify this built-in role. Doing so can create a situation in which no users can access certain functionality.

💡 The process of defining task-based roles requires in-depth knowledge of both the organization's business processes and the Acumatica ERP approach to security.

💡 For contact-related user types, you can associate only roles that are marked as guest roles (that is, they have the Guest Role check box selected) on the User Roles (SM.20.10.05) form.

Roles and Access Rights

In Acumatica ERP, you can control roles' access rights to system entities at multiple levels:

- The Suite Level

- The Module Level
- The Screen and Report Level
- The System Objects Level
    - Forms
    - Form controls

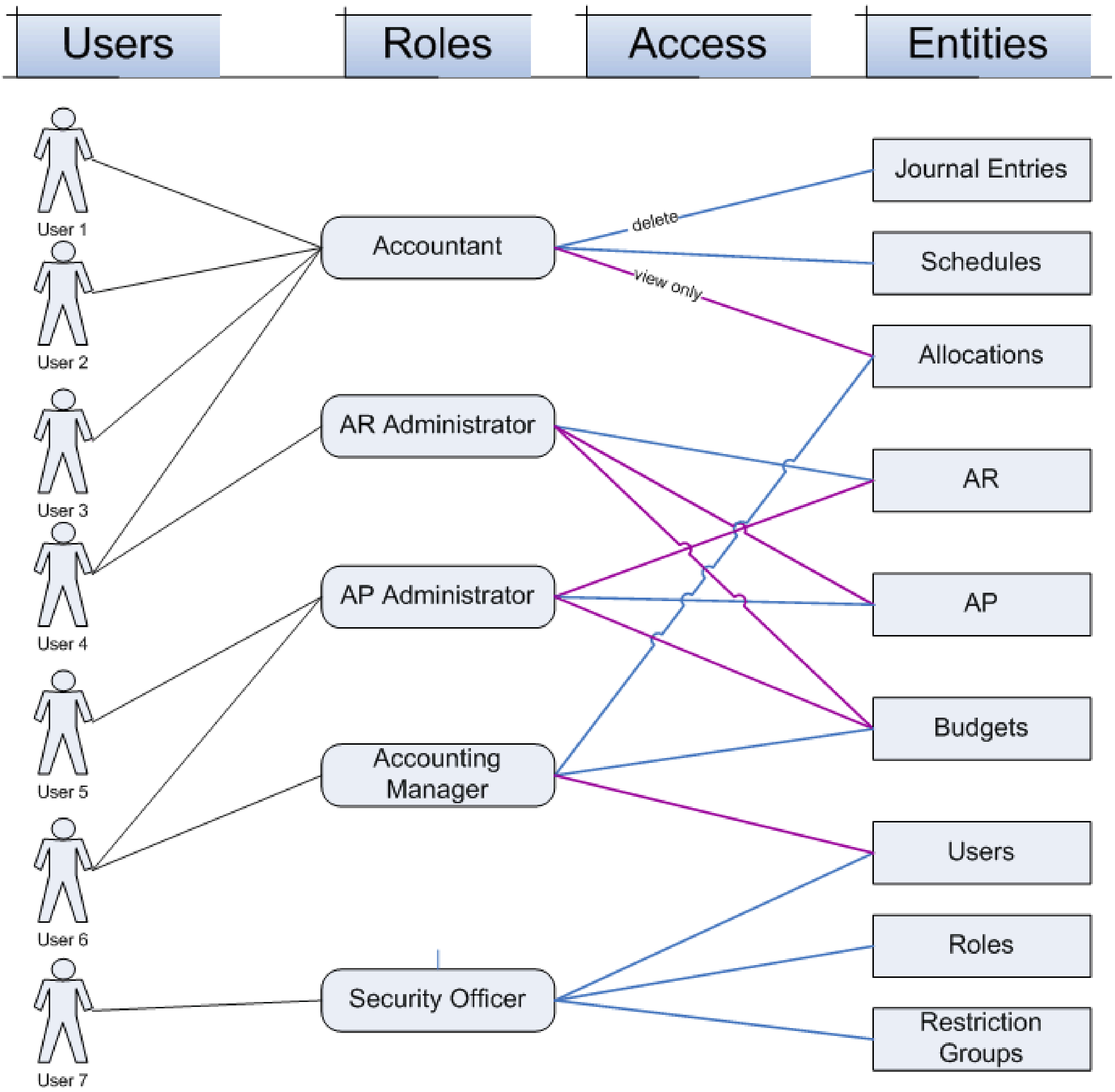
At the suite, module, and screen levels you can define 4 rights (Granted, Revoked, View Only, Not Set). At the object level you can define 6 rights (Inherited, Revoked, View Only, Edit, Insert, Delete).

You use the User Roles (SM.20.10.05) form to create roles and assign system users to roles and the Access Rights by Role (SM.20.10.25) form to configure the role's access rights.

Most roles have access to only particular modules. Among roles that have access to a module, some may have access to only a few forms within the module, while others may have access to all forms. For example, in the Accounts Payable module, users with one role may enter bills, while users with another role may approve bills for payment.

Consider an example of some roles a business might set, as shown in the illustration below:

- **Accountant role**: Allows its users full access (Delete level) to journal entries and schedules, and limited (View Only) access to allocations.
- **AR Administrator role**: Allows its users full access (Delete level) to Accounts Receivable documents; they may also view Accounts Payable documents and budgets.
- **AP Administrator role**: Allows its users full access (Delete level) to Accounts Payable documents; in addition, they may view Accounts Receivable documents and budgets.
- **Accounting Manager role**: Allows its users full access (Delete level) to allocations and budgets; they may view other user accounts too.
- **Security Officer**: Allows its users full access rights to user accounts, roles, and restriction groups.



Note that User 7 is assigned only one role, while the other users have multiple roles, in accordance with their responsibilities. A user's access rights to an entity are defined by the highest level of permission among the user's roles. With the roles shown here, if another user were given both the Accounting Manager and Accountant roles, the user's access rights to allocations would be Delete from the Accounting Manager role, rather than View Only from the Accountant role, because Delete is the higher level of permission.

Setting Up Access Rights at the Module Level

At the module (Suites, Modules, Screens, Reports) level, a role can have the following levels of access rights.

Level	Description

Granted	Allows access to the module and its forms. You can, however, limit or revoke access to particular forms within the module. The module appears on the Navigation menu with only forms to which access is higher than Revoked.
View Only	Allows restricted access to the module and its functionality. A role's *View Only* access rights to the form will allow a user with the role to view the form and any records associated with the form in drop-down lists on other forms.
Revoked	Prohibits access to the module and its forms; the module and forms do not appear on the Navigation menu. However, a role with Revoked access to the module may have access to functionality of forms within it. A user with such a role can access forms within the module only from other forms that have applicable commands; the user won't be able to view the forms on the Navigation menu.
Not set	Means that access rights to the module have not been set. This option, which is automatically set for new roles, allows access to the module until for at least one role, Revoked or Granted access rights have been set to the module. After that, this option prohibits access.

When you configure a new role, its level of access rights to all modules is by default Not Set. You can set access to a module and its objects in two ways:

- By first setting access to the module, and then adjusting the role's access to objects within the module
- By first setting access levels within the module, and then setting access to the module

If you are setting the role's access in the first way, when you change Not Set to a specific option for the module, the role's access rights to the module's forms are automatically set as follows:

- Changing to Granted for the module: Access to all the module's forms is set to Delete.
- Changing to Revoked for the module: Access to all the module's forms is set to Revoked.

This default behavior is helpful if the role's access level to most objects is Delete or Revoked.

If you use the second way of setting access for the role, then when you change the access to the node from Not Set to Revoked or Granted, the levels inside the module will remain as you have set them.

Although each module's reports are grouped under a separate node, access to them is implemented like access to the modules. A role with Revoked access rights to the node and permissive access to some of its reports will be able to access them by using actions on other forms.

If you first set access to the Reports node, the role's levels to all reports will be set automatically as Delete (if you choose Granted for the node) or Revoked (if you choose Revoked for the node). If you first set access levels to particular reports, and only then set access levels to the node, levels to reports will not be updated when you set Granted or Revoked level to the Reports node.

Set the access for specific role to a module as follows:

1. **Initial setup**: After Acumatica is installed; the module and all the forms have access set to **Not Set**. At this stage:
   - If you set the module to Granted, access for all the forms below the module level will change from Not Set to Granted.
   - If you set the module to Revoked, access for all the forms below the module level will change from Not Set to Revoked.

   This all-or-nothing operation is possible only once when you initialize the security for the module. Any further changes to module-level access do not affect the access level for the forms of the module.
2. **Ongoing maintenance**: At this stage, you can change the access for specific forms by selecting Granted or Revoked. The module-level access setting can be used only to control the access to the module as a whole. Thus, if you select Revoked, users with the role will be denied access, and if you select Granted, users with the role will be able to access only the forms with Granted set.

Setting Up Access Rights at the Form Level

Within each module, you can set the role's access rights to forms. The level of access rights to the form is inherited by the entities and records created with the help of the form. Within a module, you can set any of the following levels of access rights to forms.

Level	Description
Revoked	Denies access to the form and its functionality.
View Only	Allows restricted access to the form and its functionality. A role's *View Only* access rights to the form will allow a user with the role to view the form and any records associated with the form in drop-down lists on other forms.
Edit	Allows restricted access to the form and its functionality. The user can view the form, select objects, and edit the objects' details. The **Edit** action is available on the form toolbar.
Insert	Allows restricted access to the form and its functionality. The user can view the form, select records, edit record details, and create new records or objects of the type. The **Edit** and **Insert** actions are available on the form toolbar.
Delete	Grants complete access to the form and its functionality. This level encompasses the *View Only*, *Edit*, and *Insert* levels, and adds the ability to delete objects. The **Edit**, **Insert**, and **Delete** actions are available on the form toolbar.

Not Set	Indicates that access to the form is not set. During initial system deployment, for a new role, or for a new form, this level allows access; if you've given at least one role access rights to the object at a specific level, it prohibits access.

Setting Up Access Rights to Form Controls

Generally, a role's access rights to the form fields and actions are inherited from the role's access to the form. Thus, you should set the role's permissive level of access to the system object that supports the form functionality first. Then you can set access rights to the form controls.

If a particular form toolbar contains form-specific actions, you can prohibit the role's access to them by setting Revoked for the action. You can hide fields or make them read-only by appropriately revoking or View-only access level, be careful with this as if these fields does not have default values, forms will not function correctly.

> Configuring a role's access at the form control level requires in-depth knowledge of Acumatica functionality.

Consider following scenarios:

Scenario 1: If you need to allow a role to access only a few forms in a specific module and deny access to the rest of the forms of the module, do the following on the Access Rights by Screen (SM.20.10.20) form:

1. In the System Tree (left) pane, select the module to view different roles' access rights to it in the right pane. For the role you're working with, select Revoked. Save your changes.
2. In the System Tree pane, select a form (within this module) that you want the role to access. For the role, select Granted. Repeat this step for each form you want users of the role to access.
3. In the System Tree, select the module, and for the role, select Granted.

Scenario 2: If you need to allow a role to access all the forms in a specific module except for a few forms, proceed as follows on the Access Rights By Screen form:

1. In the System Tree pane, select the module. For the role, select Granted.
2. In the System Tree pane, select a form (within this module) to which you want to deny access. For the role, select Revoked. Repeat this step for each form to which you want to deny access.

Questions

1) What are the different ways to secure access and authorization in Acumatica ERP?

2) What is the difference between Contact and Employee Linked Entity users?

3) List password policy options available in Acumatica ERP.

4) What level of access rights available at module level?

5) What level of access rights available at forms and controls level?

Hands on – Setting up New User Role

This exercise will create a role that is designed for an employee who works in a warehouse. The person's responsibilities include creating shipments, receiving purchase orders, preparing physical inventory counts, updating ABC codes, updating movement classes, and initiating inventory transfers. As an employee of the company, the person will also have to enter timesheets and expense reports.

Create a New User Role

1. Open ***User Roles*** screen (SM.20.10.05)

    a. *Configuration > User Security > Manage > User Roles*

2. Click "+" to add a new role. Complete the fields as follows:

Screen	Field	Value
User Roles	Role Name	Shipping
User Roles	Role Description	Shipping Role
User Roles	Guest Role	{unchecked}

When complete, press ***Save***.

Assign Rights to the Newly Created Role

1. Open ***Access Rights By Role*** screen (SM.20.10.25):

    a. *Configuration > User Security > Manage > Access Rights By Role*

2. In **Role Name**, select the ***Shipping*** role we just created.

3. Create Suite level permissions.

    a. Click the company name, this will reveal all suites in right hand grid.

    b. Set access rights as follows:

Description	Access Rights
Organization	Not Set
Finance	Not Set
Distribution	Granted
Configuration	Not Set
System	Not Set
Help	Not Set
Hidden	Not Set

    c. **Save** your changes

4. Set module level permissions:

    a. Click the distribution suite, if you completed step 3 correctly, all modules should inherit the ***Granted*** permission that you assigned at the Suite level.

    b. Set access rights as follows:

Description	Access Rights
Inventory	Granted
Sales Orders	Granted
Purchase Orders	Granted
Purchase Requisitions	Revoked

    c. **Save** your changes

5. Set screen level permissions (let's assume that we don't want our shipping person to be able to change sales orders, invoices, and payments):

    a. Open the distribution suite, sales orders, Work Area, and click ***Enter***. If you completed step 4 correctly, all screens should inherit the ***Delete*** permission based on your module level entries.

    b. Set access rights as follows:

Description	Access Rights
Sales Orders	View Only
Shipments	Delete
Invoices	Delete
Payments and Applications	View Only

    c. **Save** your changes

6. Set form level permissions (let's assume that we don't want our shipping person to be able to change the shipment address):

    a. Open the distribution suite, sales orders, Work Area, Enter, and click ***Shipments***. If you completed step 5 correctly, all screens should read *Inherited*.

    b. Leave all rights as Inherited, except the one change below.

Description	Access Rights
Shipment Address	View Only

c. **Save** your changes

Hands on – Adding a new User

This exercise will create a new employee who works in a warehouse. The person's responsibilities include creating shipments, receiving purchase orders, preparing physical inventory counts, updating ABC codes, updating movement classes, and initiating inventory transfers. As an employee of the company, the person will also have to enter timesheets and expense reports.

Create a New User and Assign Roles

1. Open ***User*** screen (SM.20.10.10)

   a. Configuration > User Security > Manage > Users

2. Click "+" to add a new user. Complete the fields as follows:

Form or Tab	Field	Value
Summary Section	Username	ipick
Summary Section	Generate Password	{unchecked}
Summary Section	Password	123
Summary Section	Guest Account	{unchecked}
Summary Section	User Type	{leave blank}
Summary Section	Contact	{leave blank}
Summary Section	First Name	Ian
Summary Section	Last Name	Pick
Summary Section	Email	{your email}
Summary Section	Comment	{leave blank}
Summary Section	Allow Password Recovery	{accept default}
Summary Section	Allow Password Changes	{accept default}
Summary Section	Password Never Expires	{accept default}
Summary Section	Force User to Change Password	{unchecked}
Roles	Internal User	{checked}
Roles	Main Users	{checked}
Roles	Shipping	{checked}
Roles	Wiki Author	{checked}
Roles	All Others	{unchecked}

By assigning multiple roles, we will give our user access to fields in the employee tab as well as the distribution capabilities assigned in our shipping role and the documentation in the wiki.

When complete, press ***Save***.

3. Logout as the administrator, then login as your newly created user. Navigate in the system to verify the permissions that you established.

Add Timesheets and Expense Reports

Note that the user that we just created cannot enter timesheets or expense reports because the roles we assigned did not include those items. We could assign the employee role, but this role has access to too many items in the demo data. Instead we will create a new role and assign that to our newly created user.

4. Logout and login as the administrator

5. Create new role called Employee Forms using the ***User Roles*** (SM.20.10.05) screen.

   a. Configuration > User Security > Manage > User Roles

   b. Assign your user to this role in the membership tab

6. Open the ***Access Rights by Role*** (SM.20.10.25) screen.

7. Create Suite level permissions.

   a. Click the company name, this will reveal all suites in right hand grid.

   b. Set access rights as follows:

Description	Access Rights
Organization	Granted
Finance	Not Set
Distribution	Not Set
Configuration	Not Set
System	Not Set
Help	Not Set
Hidden	Not Set

   c. **Save** your changes

8. Set module level permissions:

   a. Click the organization suite, if you completed step 3 correctly, all modules should inherit the ***Delete or Granted*** permission that you assigned at the Suite level.

   b. Set access rights as follows:

Description	Access Rights
Communication	Delete
Customer Management	Revoked

Projects	Revoked
Time & Expenses	Granted
Organization Structure	Revoked

   c. **Save** your changes

9. For simplicity, we will not set screen level permissions (however, you may want to restrict screens like Release Expense Claims, Release Time Cards, Approve Time Activities, Release Time Activities, Release Equipment Time Cards, etc.)

10. Logout and login as Ian Pick to verify your settings.

Site Security Options

Learning Objects:

- *Learn various account and password policy options available in Acumatica ERP and how to apply them*
- *Learn access audit options available in Acumatica ERP and how to use them*

Acumatica allows you to enforce various security policies for the local and domain users, including audit policy and policies for maintaining local user accounts and passwords. You can also upload and register encryption certificates to be used for data encryption and signing document added to Acumatica by users.

With Acumatica, companies can easily implement appropriate policies for user accounts, passwords, and auditing.

Account Policy Options

Acumatica supports integration with Active Directory, so domain users can use their network logins and passwords to log in to Acumatica. With integration in effect, password and username policies for domain users are set at the domain level by using the Active Directory. For details, see Integrating Acumatica ERP with Active Directory.

Acumatica protects local users by providing policies that controls whether the users may change their passwords at will, whether they are forced to change their passwords periodically, and whether they are permitted to recover their passwords if they forget them. You can limit the number of times a user can enter an incorrect password and temporarily lock out the account.

The options listed below can be found on the Security Preferences (SM.20.10.60) and Users (SM.20.10.10) forms.

Option	Description
Force User to Change Password Each *x* Days	Sets an interval when users must change their passwords. We recommend that the users change passwords periodically, such as every 90 or 180 days. Using shorter periods may reduce security because users may find it hard to create complex and easy-to-remember passwords often. This could encourage them to write down passwords or choose simpler ones. This option is located on the Security Preferences form.
Lock Account After *x* Unsuccessful Login Attempts	Allows you to limit the number of times users may attempt to log into the system. This system-wide option is located on the Security Preferences form.

Lock Account for *x* Minutes	Allows you to lock out a user's account for some number of minutes after the user has tried unsuccessfully to log in. This system-wide option is located on the Security Preferences form.
Reset Lockout Counter After *x* Minutes	Unlocks the account, when it has been locked out, after the specified time interval. This system-wide option is located on the Security Preferences form.
Force User to Change Password on Next Login	Requires the user to change the password assigned by the system administrator on the first login.
Allow Password Changes	Allows the user to change his or her password at will.
Password Never Expires	Sets a never-expiring password for the user. Do not use this option for the majority of users; it will result in lower security for the site.
Allow Password Recovery	Allows the user to recover his or her password.

Password Policy Options

You can set system-wide password policy by using the following options on the Security Preferences form.

Field	Description
Minimal Password Length *x* Characters	The minimum password length; a minimum of 6 to 8 characters is recommended. Too-short passwords are easier to break. This system-wide option is located on the Security Preferences form.
Password Must Meet Complexity Requirements	This system-wide option is located on the Security Preferences form. The option requires that the passwords have at least three of the following four features: • Uppercase letters • Lowercase letters o Special symbols o Digits
Additional Password Validation Mask	A regular expression you can set to enforce additional regulations—for example, to exclude some special symbols not supported by third-party software (if used). If such a mask is set, type into the **Incorrect Password Alert** field the message to be displayed to users if their passwords don't meet this additional requirement. This system-wide option is located on the **Security Preferences** form.

Password Masks and Regular Expressions

In some cases, you may need to set a regular expression to enforce the company's password policies. For example, a regular expression could define valid and invalid characters, specific formats (for example, the password must begin with a letter), or minimum and maximum password lengths. Below are some examples of regular expressions and their explanations:

- ^(?=.*\d).$ - The password must be 4 to 8 characters long and include at least one numeric digit.
- ^[a-zA-Z]\w{3,9}$ - The password must be 4 to 10 characters long, the first character must be a letter, and no character is permitted except letters, numbers, and underscores.

Acumatica can provide an additional authentication option for users who forget their passwords. A user may set a Password Recovery Question and a Password Recovery Answer to be used in case he or she forgets the password. Clicking the **Forgot your password or login?** link on the login screen activates the password recovery dialog. If the user provides the correct answer, the system generates a new password and sends it to the user's email account.

Users can set parameters for password recovery on the User Profile… (SM.20.30.10) form. You set the Allow Password Recovery option on a per-user basis using the Users form (SM.20.10.10).

In Acumatica, passwords are hidden (by asterisks) on entry. They also can be stored encrypted. For password encryption, a hash or a unique encryption key is used. The key is generated during Acumatica installation. The password encryption options are located on the Security Preferences (SM.20.10.60) form.

Option	**Description**
Clear	User passwords are stored as they are, not encrypted. This is not a recommended option because it increases the system vulnerability.
Hash	The system converts each password to a *hash value* and stores the hash value, not the password. When a user logs in, the system compares the hash value of the input with the hash value stored in the database. This type of encryption is irreversible.
Encrypted	Passwords are encrypted using a unique encryption key generated during Acumatica installation, with the use of the web.config file.

Site Security and Access Audit Options

A day-to-day operation in Acumatica creates a log to record all the events that might signal security problems. Logs are stored in the system for the period of time you specify. On the Security Preferences form (SM.20.10.60), you can choose which events will be audited from the following events:

- User login and logout

- Failed login
- Screens accessed
- Expired sessions
- Successful email sending
- Failed email sending

Field-Level Auditing

The development of automatic data processing has made it necessary to consider protecting sensitive information. In certain highly regulated industries, companies must implement auditing to address identity-management concerns related to compliance issues. Regulations such as Sarbanes-Oxley (SOX), the Health Insurance Portability and Accountability Act (HIPAA), and the Payment Card Industry Data Security Standard (PCI DSS) all have extensive requirements on auditing user identity and access to system resources.

If your company must comply with such regulations, wants to ensure accountability and the ability to track user actions in the system, or wants to implement sound auditing procedures, you can use the field-level auditing available in Acumatica ERP.

Field-level auditing functionality gives you the ability to monitor and record user actions on Acumatica ERP forms as recorded in the system. The audit trail holds records of every change users have made on the monitored forms, such as changes to documents and their properties, modifications to customer accounts or employee records, and changes in security policies. You can also see who made the changes and when they took place.

Managing Access to Field-Level Audit Feature

Before you start configuring and turning on auditing for specific forms, you must enable the Field-Level Audit feature. To give users the ability to configure, turn on, and turn off auditing of forms, as well as view the audit trails, you assign them the *Field-Level Audit* role and give them access to the forms used to manage field-level auditing.

Configuring and Turning on Auditing of a Form

If you want to maintain a record of user activity on any form, you can configure and turn on auditing of the form. Granular auditing is configured, turned on, and turned off on a per-form basis on the Audit screen (SM.20.55.10). After the form audit is configured, you can quickly turn on and off auditing of the form. When you turn on the audit, every time a user makes changes to a document associated with the form and clicks Save, a record is added to the audit trail the system maintains for the form. This record contains the details of the modification, including who modified the document, what changes were made, and when the changes occurred.

For more granular control, you can perform auditing of individual database tables associated with the forms and specific database table fields. You can audit entire database tables or only certain database table fields, such as the fields associated with the interface elements including sensitive encrypted fields on the form. These sensitive fields are logged with changed values and are displayed as asterisk.

Viewing an Audit Trail

When auditing is turned on for a form, you can select a document and view the changes made to the document directly from the form by clicking Audit History on the Help menu. The audit trail shows who modified the document and when, what form was used, when the modification took place, and what changes were made.

Also, you can view all the changes made in the database tables of the audited form. The audit trail shows the user who modified the document, the date and time of the update, and the details of the modification of the selected database table. You can filter the modifications that you view in any audited database table by user and by date range.

Questions

1. List and discuss various account policies available in Acumatica ERP
2. List and discuss various password policies available in Acumatica ERP
3. What audit options are available in Acumatica ERP?
4. Set password policy such that
   a. Password is at least 6 characters long
   b. Should not start with a number
   c. And should contain at least one number and special symbol
5. Ensure Users passwords are encrypted in the database table.

Hands on – Setting Field Level Auditing

This exercise will set field level auditing to keep a record of changes that are make to a sales order.

Verify System Settings

1. Open ***Enable/Disable Features*** screen (CS.10.00.00)
    a. *Configuration > Common Settings > Licensing > Enable/Disable Features*
2. Verify that ***Misc > Field-Level Audit*** is checked
    a. If not checked, click the ***Modify*** button at the top of the screen, check the box, and then press the ***Activate*** button.

Create a new Auditing Feature

3. Open ***Audit*** screen (SM.20.55.10)
    a. *System > Management > Manage > Audit*
4. Select a screen to Audit
    a. Click *Distribution > Sales Orders > Work Area > Enter > Sales Orders*
    b. After making the selection, the system will load the schema for the screen. This will include tables and fields that can be audited. The table SOOrder will automatically be selected as "active" – we will not make any changes to this.
    c. Set the ***Description*** Field to "Sales Order Auditing"
5. **Save** your changes

Create Transactions and Perform an Audit

1. Navigate to Sales Order screen and create a sales order

Form or Tab	Field	Value
Summary Section	Order Type	SO
Summary Section	Customer	ABARTENDE
Summary Section	Description	Sales Order with Audit
Summary Section	All other fields	{accept default}
Document Details	Branch	MAIN
Document Details	Inventory ID	Z000L96065
Document Details	Subitem	0-0
Document Details	Warehouse	WHOLESALE

Document Details	UOM	{accept default}
Document Details	Quantity	1
Document Details	Unit Price	10
All Others	All others	{accept default}

2. **Save** your changes.

3. Logout as administrator and login as Clark Willard

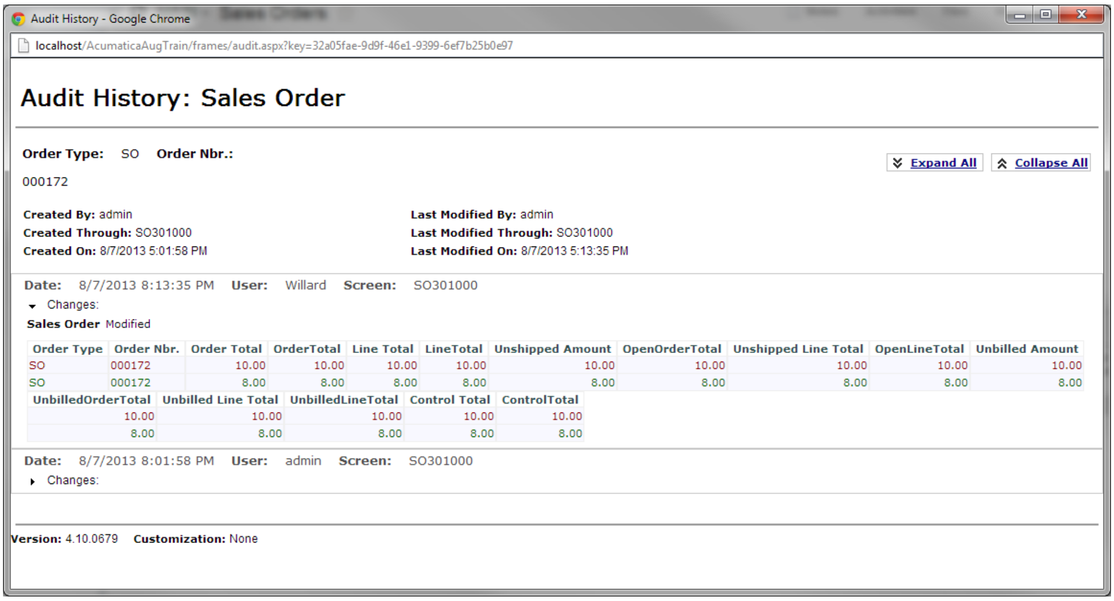
   a. Username: Willard
   b. Password: most likely will be setup (you will have to change it)
   c. **Note**: you might have to add the sales order screen permission to the “Sales” role on screen ***Access Rights By Role*** (SM.20.10.25)

4. Open ***Sales Orders*** screen

   a. Select the sales order you just created as the administrator
   b. Change the ***Unit Price*** from 10 to 8 in the ***Document Details*** tab.
   c. **Save** your change

5. Logout and log back in as the administrator

6. Open ***Sales Orders*** screen
   a. Select the sales order that you have been working with.
   b. Select *Help > Audit History* … in the upper right corner.
   c. Verify that your screen is similar to the one pictured below.


Audit History - Google Chrome
localhost/AcumaticaAugTrain/frames/audit.aspx?key=32a05fae-9d9f-46e1-9399-6ef7b25b0e97
Expand All
Collapse All

Audit History: Sales Order

Order Type: SO **Order Nbr.:** 000172

Created By: admin
Created Through: SO301000
Created On: 8/7/2013 5:01:58 PM

Last Modified By: admin
Last Modified Through: SO301000
Last Modified On: 8/7/2013 5:13:35 PM

Date: 8/7/2013 8:13:35 PM **User:** Willard **Screen:** SO301000
▾ Changes:
Sales Order Modified

Order Type	Order Nbr.	Order Total	OrderTotal	Line Total	LineTotal	Unshipped Amount	OpenOrderTotal	Unshipped Line Total	OpenLineTotal	Unbilled Amount
SO	000172	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00
SO	000172	8.00	8.00	8.00	8.00	8.00	8.00	8.00	8.00	8.00

UnbilledOrderTotal	Unbilled Line Total	UnbilledLineTotal	Control Total	ControlTotal
10.00	10.00	10.00	10.00	10.00
8.00	8.00	8.00	8.00	8.00

Date: 8/7/2013 8:01:58 PM **User:** admin **Screen:** SO301000
▸ Changes:

Version: 4.10.0679 **Customization:** None

Integrating Acumatica ERP with Active Directory

Learning Objects:

- *Learn to integrate Active Directory authentication with Acumatica ERP*

Acumatica supports integration with Active Directory (AD). This integration allows companies to maintain centralized account and password policies at the domain level. Thus, domain users can log on to Acumatica using their domain credentials, and user access rights in Acumatica are applied automatically based on the predefined mapping rules between AD groups and Acumatica roles.

To integrate an instance of Acumatica ERP with Active Directory, you take the following actions, each of which is described in a section below:

1. Enable Active Directory integration by modifying the web.config file of the application instance.
2. Map the roles configured in Acumatica to the groups configured in the Active Directory domain via the User Roles (SM.20.10.05) form in Acumatica.

Enabling Active Directory Integration

To enable Active Directory integration, do the following:

1. Create an Active Directory user account that has *Read* rights throughout the entire AD forest. This user account must have at least *Read* rights to the following properties defined in the Active Directory Schema: **objectSid**, **distinguishedName**, **sAMAccountName**, **displayName**, **description**, **lastLogon**, **pwdLastSet**, **primaryGroupID**, and **memberOf**.
2. Add the following section to the Web.config file of the application instance that you want to integrate with Active Directory:

   ```
   <system.web>
   <activeDirectory enabled="true" path="domain_path" dc="domain_name"
   user="user_name" password="user_password" />
   </system.web>
   ```

   The path=”domain_path” parameter is required and should be your AD server path.

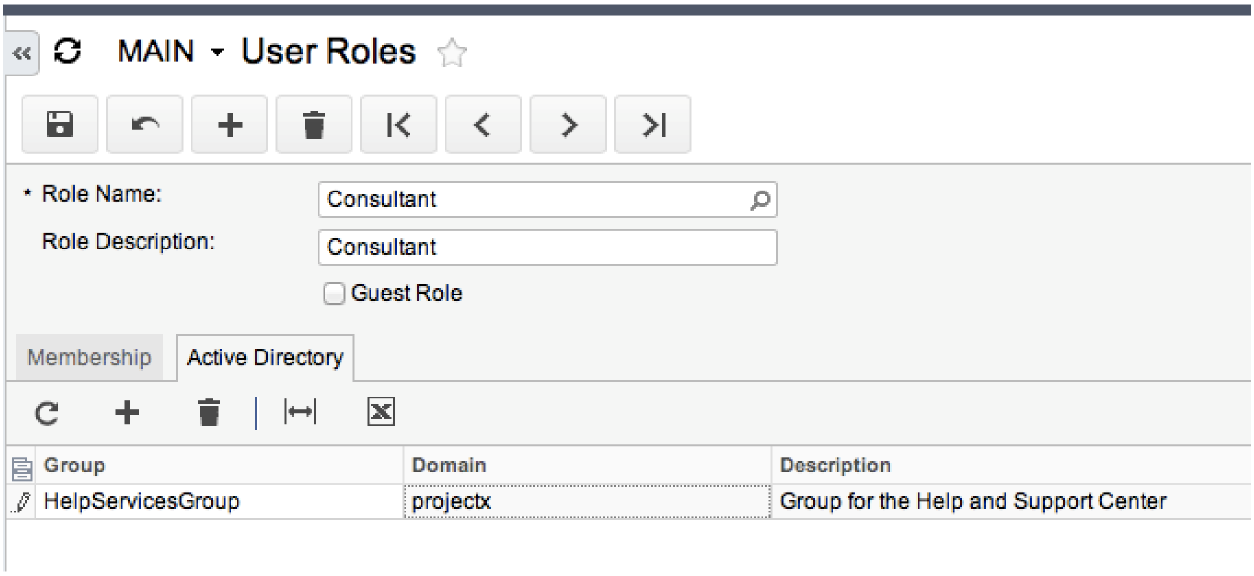
   The dc="domain_name" parameter is optional and required only in the case if you have more than one domain in the forest. The user account credentials belong to the user account you have created in Step 1.

⚠ If you integrate Acumatica with Active Directory Federation Services, you must also configure claims-aware authentication.

Mapping Active Directory Groups to Roles in Acumatica ERP

After you enable Active Directory integration, you need to map Active Directory groups to user roles defined in Acumatica ERP using the User Roles (SM.20.10.05) form. Do the following for each role you want to associate with Active Directory groups:

1. Navigate to **Configuration > User Security > Manage > User Roles**.
2. In the **Role Name** field, select the role you want associate with one or more Active Directory groups.
3. On the **Active Directory** tab, click **New Line**.
4. In the **Group** column, click the selector button and select the Active Directory group to associate with the role (see the screenshot below).
5. Repeat the steps 3 through 4 multiple times to add all the required Active Directory groups.
6. Click **Save**.



* Role Name: Consultant

Role Description: Consultant

☐ Guest Role

Group	Domain	Description
HelpServicesGroup	projectx	Group for the Help and Support Center

Enabling Active Directory integration does not affect the standard authorization and authentication mechanism of Acumatica. With the Active Directory integration enabled, you still can create regular (non-AD) users in Acumatica.

After you have completed the above steps, domain users can log on to Acumatica using their domain credentials, as shown in the following example:

```
Login: <domain name>\<user name>
Password: <user password>
```

When a domain user enters his domain credentials on the Acumatica login screen, the following user authentication mechanism is implemented:

1. The application instance sends an authentication request to the AD server to validate the user's credentials.
2. Upon successful validation, Acumatica requests the AD server for the list of the user roles defined in the Active Directory.
3. Acumatica compares the list of AD groups with the internal Acumatica roles based on the mapping rules defined in Acumatica.
4. If at least one Acumatica role is found that is associated with an AD group to which the domain user account is assigned, then the user is granted the right to log on to Acumatica.
5. Acumatica then defines the user's access rights within the application instance based on the internal list of roles.

Questions

1. What are the benefits of AD integration in Acumatica ERP?

2. Locate AD settings in the web.config and enable it.

3. Discuss how AD groups and Acumatica roles work together to provide user access and

   authentication in Acumatica ERP

4. Discuss the effects of turning off AD integration at a later date.

Digital Certificates

Learning Objects:

- *Learn how to:*
    - *Protect sensitive information in database*
    - *PDF signing*

Acumatica uses digital certificates to store sensitive information in the database encrypted and to authenticate documents (PDF files) shared or sent electronically. Digital certificates are electronic credentials that bind the identity of the certificate owner to a pair of electronic keys (a public key and a private key) that can be used to encrypt the data and to digitally sign documents. These certificates can be purchased from a recognized certification authority. Each certificate has a password that is used to validate the owner of the certificate in case you need to reinstall the system or move the database.

Uploading of Certificates

Digital certificates used by Acumatica for database encryption and for signing documents have the .pfx extension.

Before you can upload digital certificates to the system, add the .pfx extension to the list of allowed extensions on the File Upload Preferences (SM.20.25.50) form.

To use a certificate of either type in Acumatica, register it on the Encryption Certificates (SM.20.05.30) form in the following way:

1. Specify the certificate name in the **Name** field.
2. Enter the password for the certificate.
3. Click the paper clip icon at the beginning of the row, and click **Add File** to open the **File Upload** dialog.
4. In the dialog, make sure the **Upload a new file** option is selected. Browse your local computer or network to locate the certificate file. Click **Upload**.

The certificate file will be attached to the certificate registration record.

Database Encryption

Acumatica requires a security certificate to encrypt sensitive information, such as credit card numbers. If needed, the certificate already used in the system may be replaced by a new one.

To maintain the Acumatica database encrypted, use the Certificate Replacement (SM.20.05.35) form.

To encrypt the database, perform the following steps:

1. Select a certificate whose key will be used for database encryption. You can select only from the certificates uploaded to the website.
2. View the certificate currently used for database encryption in the Current Certificate field. If the field is blank, encryption has never been performed.
3. Click Replace Certificate on the form toolbar. This initiates the process of decrypting the data with the key associated with the previous certificate and encrypting it using the new key. Alternatively, assign the process of replacing the certificate to a schedule by using the Schedule menu on the form toolbar.

Once the process of replacing the certificate has been completed, you can remove the "old" certificate using the Encryption Certificates form.

PDF Signing

Another type of certificate is used to sign PDF files generated in the system. A PDF certificate protects the document's authenticity throughout its life cycle. For example, when the company emails its digitally signed quarterly financial statements, the recipients of the documents can be sure of the identity of the sender, and that the financial information has not been altered.

The default certificate for signing PDF files is specified on the Security Preferences (SM.20.10.60) form; this certificate is used unless users do not specify their personal certificates. Users responsible for preparing and generating such documents may use personal certificates that are specified for each user on the My Settings (SM.20.30.10) form.

Questions

1. How data in Acumatica ERP is protected?

2. What types of certificates are used in Acumatica ERP to protect data?

3. Ensure Users passwords are encrypted in the database table using certificates.

4. Enable PDF signing in Acumatica ERP.

5. Apply certificate to protect sensitive information stored in the database.

Security Forms Reference

User Roles

You can use the User Roles form to create new roles and assign roles to users. For an existing role, you can view the list of users assigned to it. If your system is integrated with Active Directory, you can map the roles configured in Acumatica to the groups configured in the Active Directory domain.

A role is a set of access rights to specific modules or other system entities. Some users are assigned only one role, while others are assigned several roles at once in accordance with multiple sets of employee responsibilities. A guest role is a role, generally given to an external user, which you configure to give restricted access to the website and to only particular modules.

Users

With the Users form, you can add users to the system and assign them roles; also, you can edit user information and delete users.

To get access to the system, users must authenticate themselves by username and password. These users should have roles assigned before they obtain system access. Each role is a set of access rights to, or permissions to work with, the system entities. Some users are assigned only one role, while others are assigned several roles in accordance with multiple sets of employee responsibilities.

If your system is integrated with Active Directory, all domain users access Acumatica by using the same credentials they use to log into local network. The password and user account policies are set at the domain level and settings in Acumatica do not affect user accounts.

Users Types

You use this form to define user types, which are used to provide default settings for creating new users. On the form, you can also define the set of roles that are available for the user of the type and the default roles to be assigned when a user of this type is created. You can add new user types, view existing user types and modify their settings, and delete unused types from the system.

Your company may need to give your partners limited access to Acumatica ERP—for instance, to facilitate the entering of contacts or customer orders. In such a scenario, you may need some way to segregate users that are internal to your company from external users, or even to give your partners means to create and manage their own users in Acumatica ERP, thus freeing your administrators of the requirement to manage these users. Your company's policy might require a restricted set of roles assigned to specific groups of these users. You can address all these requirements with the user types functionality in Acumatica ERP.

A user type is a classification of users intended for the following purposes:

- Segregating users based on their relationship to the company (internal versus external)

- Delegating the rights to create users to external users—for instance, to external administrators of Self-Service Portal
- Automatically assigning and restricting the collection of roles that are given to a new user
- Defining default setting for new users of the type

Access Rights by Screen

By using the Access Rights by Screen form, you can view and modify the access rights of roles to system modules and forms.

The following options are available for setting up access to modules:

- **Not Set**: The option is set automatically during initial system implementation or for a new role. Access to the module is allowed for all roles until for at least one role, the rights are changed to any other option. After that, for all other roles, access is denied. For a new, just created role, access to all objects is Not Set.
- **Revoked**: Access to the module is not allowed, while access to some system graphs supporting forms may be allowed. In this case, the module and all its forms will not appear in the Navigation menu, but the forms access to which is allowed may be accessed by direct links. If, for a new role, you change its access rights to a module from Not Set to Revoked, this sets the role access rights to all objects in the module to Revoked.
- **Granted**: Access to the module is allowed. Access to particular system objects of the module may be set differently. If, for a new role, you change its access rights to a module from Not Set to Granted, this sets the role access rights to all objects in the module at the Delete level.

The following options categorize the access to the lower-level objects:

- **Not Set**: Access rights are not defined, and access is allowed until another option is selected for any role. After that, access is denied.
- **Revoked**: Access to the object (and to the data created using the object) is denied.
- **View Only**: View-only access is allowed; the role cannot create, edit, or delete data using the object.
- **Edit**: The role allows the user to edit the data using the object.
- **Insert**: The role can create and edit the data using the object.
- **Delete**: The role is granted complete access rights to the object and the data created using the object.

Consider the following tips as you determine which steps to take in which order:

- If you set the access rights of a new role to the module first, the role's access levels to the graphs within the module will be set automatically to the default values based on what you have set for the module. If you select the Granted option for the module, access to all objects will be set at the Delete level. If you select the Revoked level of access to the module, access to all objects will be set at the Revoked level. This will save you time if the new role should have Delete or Revoked levels of access rights to most of the system graphs within the module; you'll need to change only levels for a few objects.

- You can set a role's access rights for system objects within the module first, and then choose a level of access rights for the module. In this case, the access rights to the module will not change the access rights to system objects within the module.
- If you want to restrict the role's access to form controls within the system graph that supports the form, you should set the level of access to the graph first. Only then may you set access to form controls.

Access Rights by Role

By using the Access Rights by Role form, you can fine-tune each role's access rights to system modules and objects. You can control access rights down to the level of specific form fields and actions. Also, you can create a role "on the fly" and configure its access to system entities. Access rights to modules can also be set on the Access Rights by Screen (SM.20.10.20) form.

The following options are available for setting up access to modules:

- **Not Set**: The option is set automatically during system implementation. Access to the module is allowed for all roles until for at least for one role, the rights are changed to any other option. After that, for all other roles, access is denied.
- **Revoked**: Access to the module is not allowed, but access to some system objects of the module may be allowed. The module and its forms will not be displayed in the Navigation menu, but the forms can be accessed through direct links from forms in other modules; they will open as pop-up forms. The module icon will not appear in the list of modules available for the role.
- **Granted**: Access to the module is allowed. Access to particular system objects of the module may be set differently.

The following options are available for access to lower-level objects:

- **Not Set**: Access rights are not defined, and access is allowed until for any role another option is selected. After that, access is denied.
- **Revoked**: Access to the system object (and to the data created using the object) is denied.
- **View Only**: View-only access is allowed; the role cannot create, edit, or delete data using the object.
- **Edit**: The role can edit the data using the object.
- **Insert**: The role can create and edit the data using the object.
- **Delete**: The role is granted complete access rights to the object and the data created using the system object.

Audit History

By using this form, you can view the log of such user activities in the system as logging in, logging out, accessing specific forms, publishing customizations, and so forth.

Certificate Replacement

Use this form to encrypt the database data using the specified certificate or to replace the certificate for a new one and to perform database encryption based on the new certificate.

Security Preferences

Use the Security Preferences form to define security settings for your organization, such as the system's password and the user account lockout policies, encryption certificates, and audit settings.

Encryption Certificates

By using the Encryption Certificates form, you can register a new certificate with the system and upload the certificate file to the database.

Acumatica requires security certificates to digitally sign PDF files generated in the system and to encode sensitive information stored in the database. A password should be provided for each certificate. Passwords are used to validate the owners of the certificate in case the system is reinstalled or you move the data to other storage, such as another computer.

Reports

The reports available in the User Security module provide information that can be useful for employees who manage users and roles.

The User Security module includes the following reports that pertain to users and roles:

- User List (SM.65.05.00): Displays the existing user accounts and the properties of the accounts in summary or detail format.
- Role List (SM.65.10.00): Lists the roles available in the system and shows how each role is populated.
- Access Rights by Screen (SM.65.17.00): Lists the system graphs available in the system and, for each, displays the roles' access rights to it.
- Access Rights by Role (SM.65.15.00): Lists the roles available in the system; for each role, displays its access rights to the system graphs. A system graph is an object that supports one or more forms and their functionality.

Row-Level Security

Learning Objects:

- *What are Restriction Groups, and*
- *How to use Restriction Groups to provide row-level security in Acumatica ERP*

The Row-Level Security module provides functionality you can use to administer user access to information and set up various security policies. Acumatica ERP lets you restrict user access to particular vendor and customer accounts, General Ledger accounts, subaccounts, subaccount segment values, budget articles, and other objects. You can create any number of restriction groups that include users and some of the most sensitive objects. Users assigned to one restriction group cannot view the objects assigned to another restriction group.

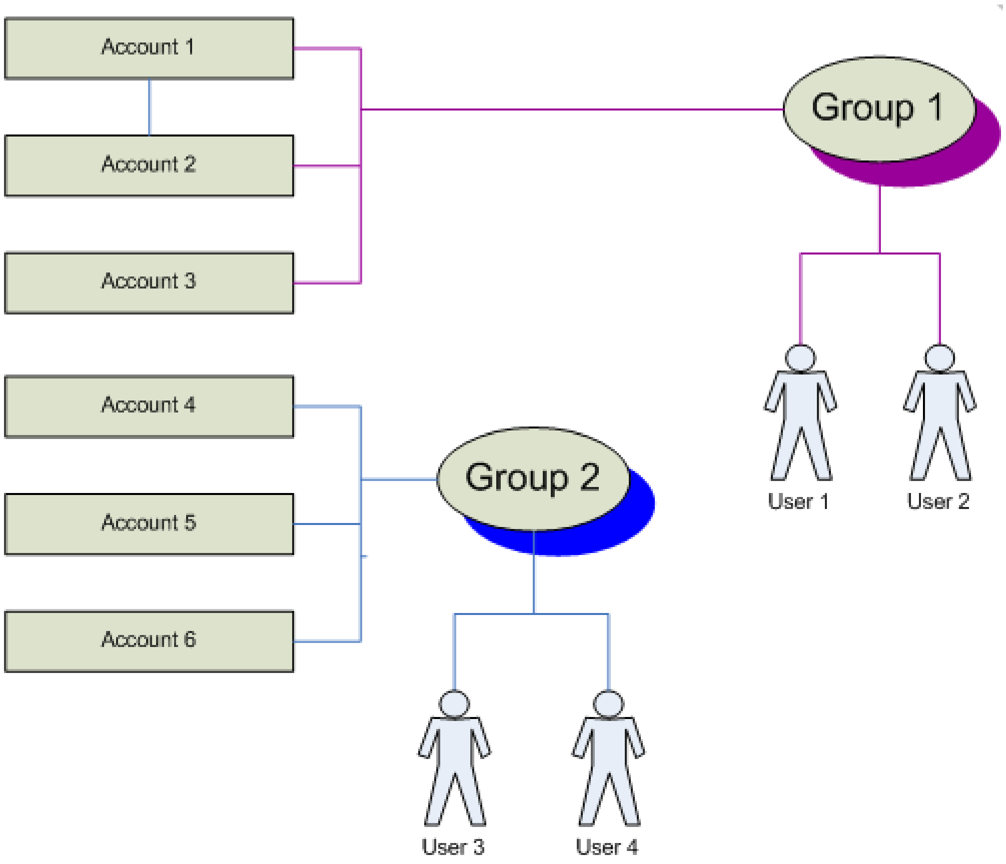
An Acumatica role can either grant its users access to all General Ledger accounts or all budget articles or other kinds of entities or deny access to any of them; it cannot be used to selectively allow access. Restriction groups with users as members are typically used to grant or deny access to specific entities selectively.

Restriction Groups

To let organizations implement more complex security requirements, Acumatica ERP provides restriction groups. A restriction group is a set of entities of two or more types that lets you accomplish one of these goals:

- If the group includes users, to grant access to entities that are members of the group to specific users only and deny access to all other users
- If the group does not include users, to relate the group's entities in a way that limits their use

Consider a typical case with restriction groups that include users and General Ledger accounts. Suppose the role allows all its users to access all General Ledger accounts, but for two groups of accounts, you want to give access to only particular users.



The illustration above shows how restriction groups can address such security needs. You define Group 1 (of type A) as a restriction group that includes only appropriate accountants (User 1 and User 2) and accounts (1, 2, and 3). Similarly, you create Group 2, which includes User 3 and User 4, as well as the accounts they should have access to (4, 5, and 6).

Among all users in the system, only User 1 and User 2 will have access to the first group of sensitive accounts (1, 2, and 3), and only User 3 and User 4 will have access to the second group of sensitive accounts (4, 5, and 6). Accounts included in either restriction group may be seen only by users who are members of the group. Users who are not assigned to any restriction group will not see the accounts associated with either group.

To allow a user in a higher position, such as the CFO, to access all these accounts, include this user in Group 1 and Group 2.

If a restriction group does not include any users, all users may view the entities that are members of the group if they are permitted by their roles, but entities included in the group become related in a way that limits their use. For example, you could create a group that includes several General Ledger accounts and several subaccounts. If on a data entry form, a user selects a General Ledger account that does not belong to the group; he or she cannot select a subaccount that does belong to the group. However, if the user selects an account that is in the group, this account can be used with the same-group subaccounts (as well as with any other subaccount not included in any restriction group).

Types of Restriction Groups

In Acumatica ERP, there are two basic types of restriction groups, A and B. Groups of both types function similarly as long as multiple groups (of the same type) do not include identical entities.

> The behavior of groups of type A and groups of type B differs slightly. Avoid mixing groups of different types created for the same kind of entities. For instance, if you create multiple restriction groups to control access to cash accounts, use groups of only one type, A or B. Because the groups of type A behave more conventionally, we recommend that you create new groups of type A. Type A is the default type of groups on all forms where you can create restriction groups.

Acumatica ERP also offers two types of inverse restriction groups: A Inverse and B Inverse. While regular restriction groups of both types are designed to allow users to access entities or entities to be used together, inverse restriction groups are generally designed to prevent users from using specific entities or entities from being used with one another.

Type A Restriction Groups

Type A is the default type of restriction group on all forms where restriction groups can be created. If a group of type A includes users and entities of a specific sort, it grants access to these entities for only users who are members of the group; access is denied for all other users. You can create multiple groups of type A, each of which grants different users access to the same entity. A user, to get access to a particular entity, should be included in at least one of these groups.

Groups of type A have been available since Version 3.0.

A restriction group of the A Inverse type denies user access to the entities included in the group. Users who are members of the group may not get access to the entities included in the same restriction group, while users who are not assigned to this group have access to the entities. If multiple groups include the same entity, access to this entity is granted to all users, whether or not they are members of the groups.

Type B Restriction Groups

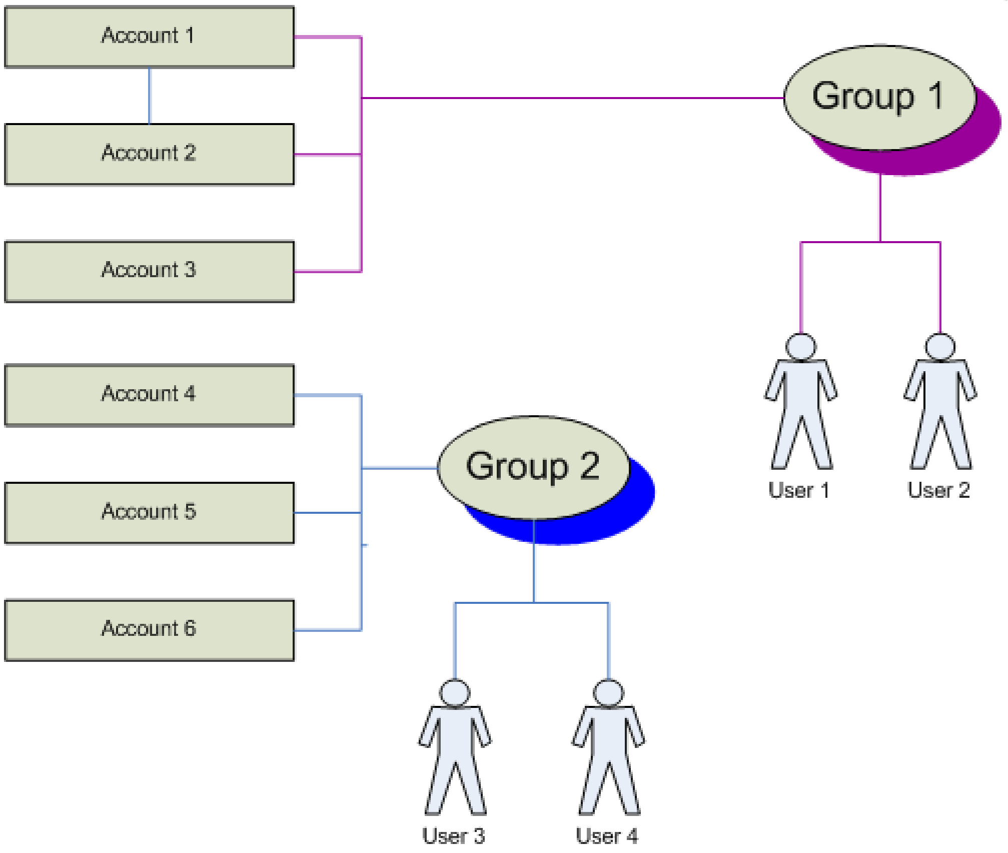
A group of type B that includes users and entities of a specific type grants access to these entities for only users who are members of the group; access is denied for all other users.

If you are using groups of type B, do not create multiple groups with the same entity. If you do, access to the entity will be denied for all users who are members of different groups that include this entity. With groups of the B type, divide entities between groups. To give a user access to entities included in different groups, include this user in these groups instead of creating a new group that includes this user and entities already included in existing groups.

Groups of type B were used in earlier versions of Acumatica as well as current ones.

A restriction group of the B Inverse type, as with a group of the A Inverse type, doesn't allow users who are members of the group to access the entities included in the same restriction group; other users who are not assigned to this group can access the entities. If multiple groups have the same entity included, access to this entity is denied for all users who are members of the groups; this is the difference between B Inverse and A Inverse groups. However, access to this entity is granted for users not included in any group.

To understand the similarities and differences in the behavior of groups of types *A* and *B*, consider the following example:



For six cash accounts, access should be granted to only trusted employees. Imagine that we create two groups of the A type:

- Group 1: Two accountants (User 1 and User 2) and the cash accounts 1, 2, and 3
- Group 2: Two accountants (User 3 and User 4) and the cash accounts 4, 5, and 6

User 1 and User 2 can perform operations with Account 1, Account 2, and Account 3, but they have no access to Account 4, Account 5, and Account 6.

User 3 and User 4 can perform operations with Account 4, Account 5, and Account 6, but these users have no access to Account 1, Account 2, and Account 3.

All other users will not see those accounts (1, 2, 3, 4, 5, and 6) on the chart of accounts and in the lookup fields on all forms to which they have access based on their roles.

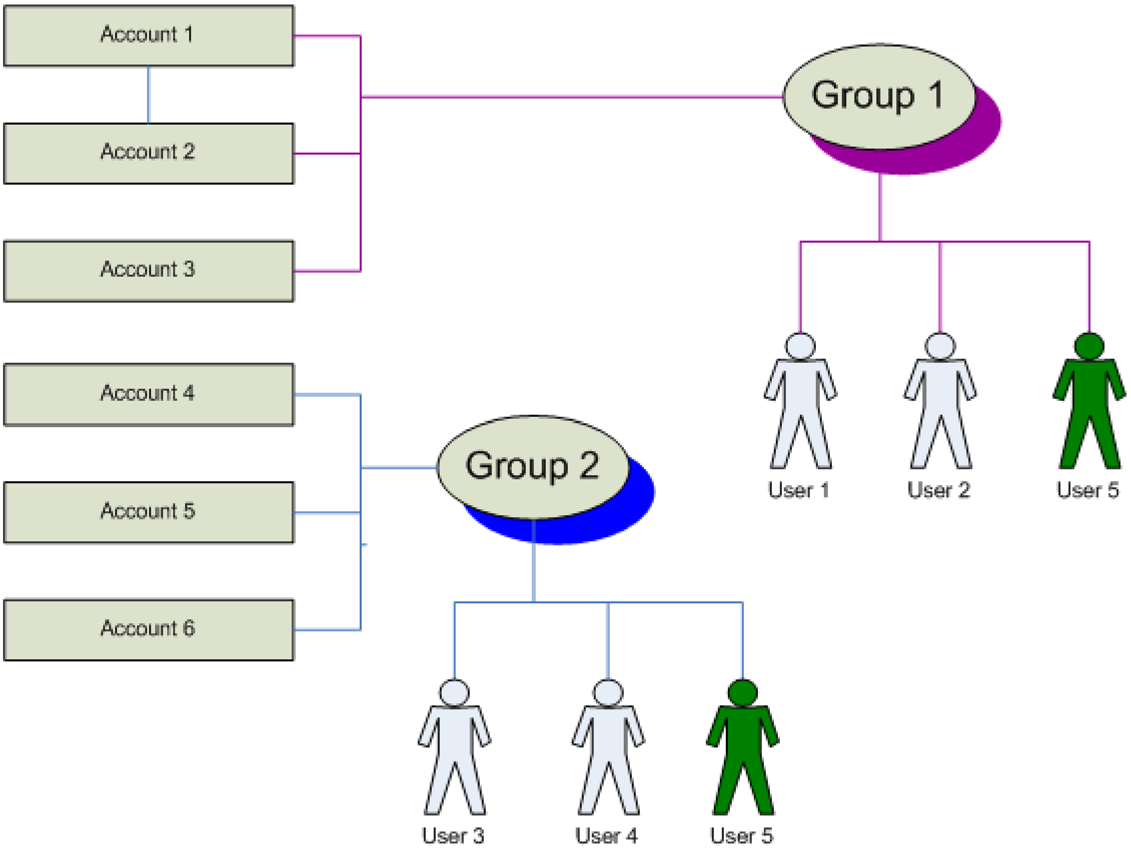
If we instead define both groups as type B, they grant access to the accounts in this example exactly as groups of type A did.

Both groups in this example include users and accounts, and the sets of accounts in Group 1 and Group 2 do not intersect. The differences between groups of types A and B appear if there are multiple groups with intersecting sets of entities. Imagine that, for instance, we need to give a new user (User 5, an accounting manager who controls cash flows) access to Accounts 1, 2, 3, 4, 5, and 6. The two ways to give User 5 access to these accounts will be:

1. Adding a New User to Existing Groups, or
2. Adding a New Group

Let's detail each one to see how this is done.

Usage Example: Adding a New User to Existing Groups

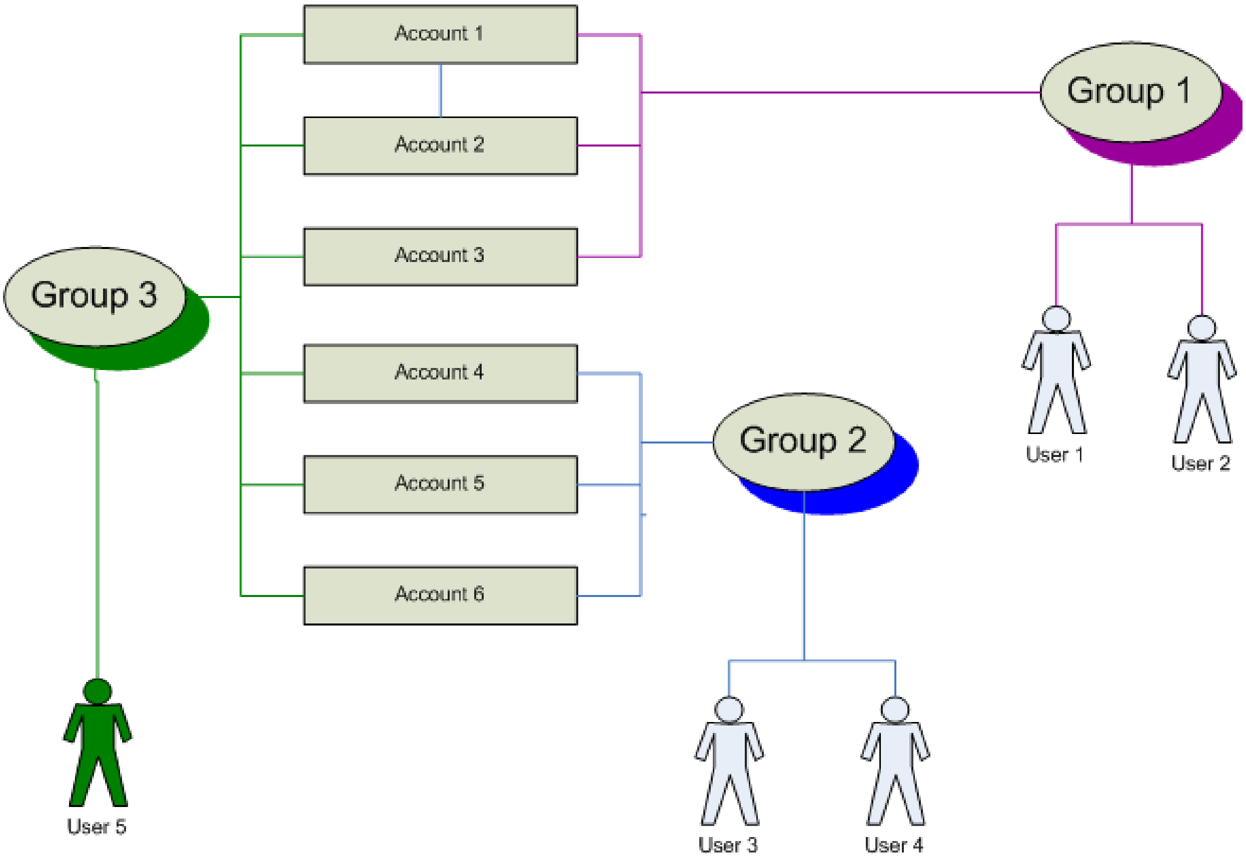


You can include User 5 in both existing groups, as shown in the illustration above. Group 1 will include User 1, User 2, and User 5, and Group 2 will include User 5 in addition to User 3 and User 4.

Whether both groups are of type A or B, they behave similarly because the groups include completely different sets of accounts. As a member of Group 1, User 5 will have access to Account 1, Account 2, and Account 3. As a member of Group 2, User 5 will have access to Account 4, Account 5, and Account 6.

If each group includes multiple entities, this is the easiest way to give new users access to these entities.

Usage Example: Adding a New Group



You can instead create a new group to grant User 5 access to Accounts 1 through 6. Include User 5 into this new group and add all six accounts, as illustrated above. In this case, groups of types A and B behave differently:

- A type: User 5 will get access to all accounts, and Users 1, 2, 3, and 4 will have access as they had before the new group was added.
- B type: Creating one more group won't work with this type. If you added the group with type B, no user would have access to any of these cash accounts. User 1 and User 2 would not get access to Accounts 1 through 3, since these accounts are also restricted by Group 3. The new group will deny access to Accounts 4 through 6 for User 3 and User 4. Meanwhile, User 5 would get no access to Accounts 1 through 3 because these accounts are members of Group 1, and the user would get no access to Accounts 4 through 6 because they are members of Group 2.

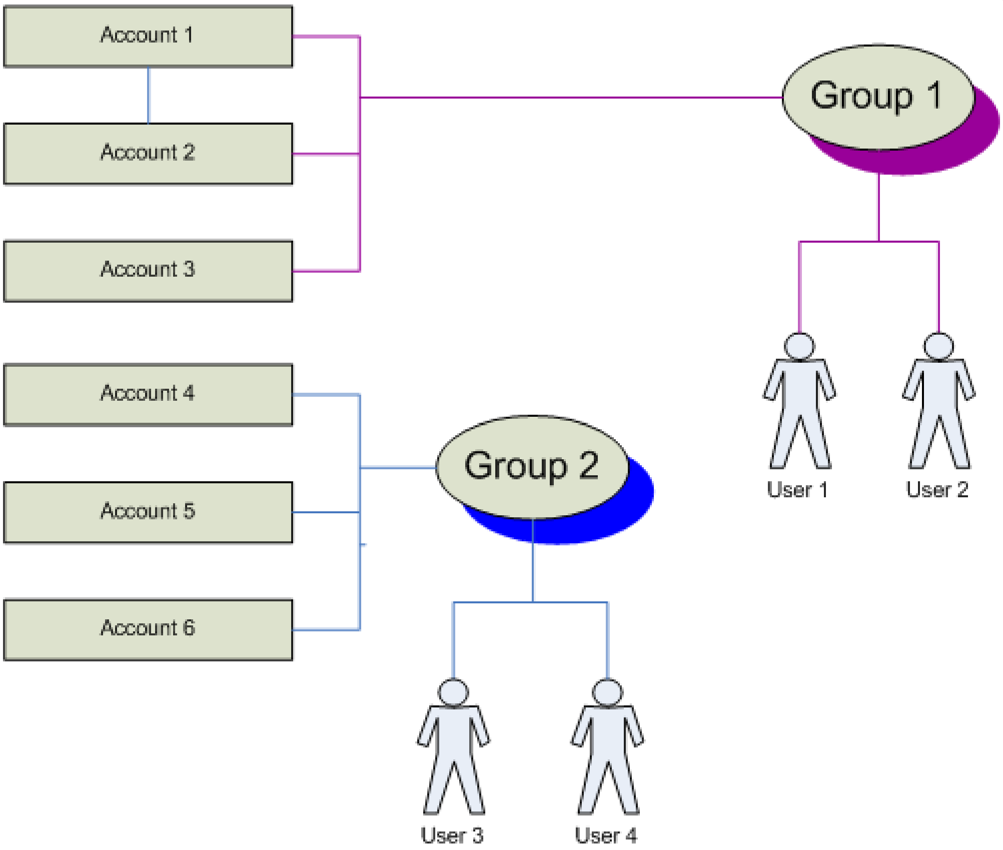
If you will use multiple groups to control user access to entities of a specific type, we recommend that you use groups of only one type, A or B.

Types A Inverse and B Inverse

Before you define inverse groups of either type, be sure you understand how each type works and plan the groups carefully. With inverse groups, users who are not members of any restriction group have access to all entities, including those included in restriction groups.

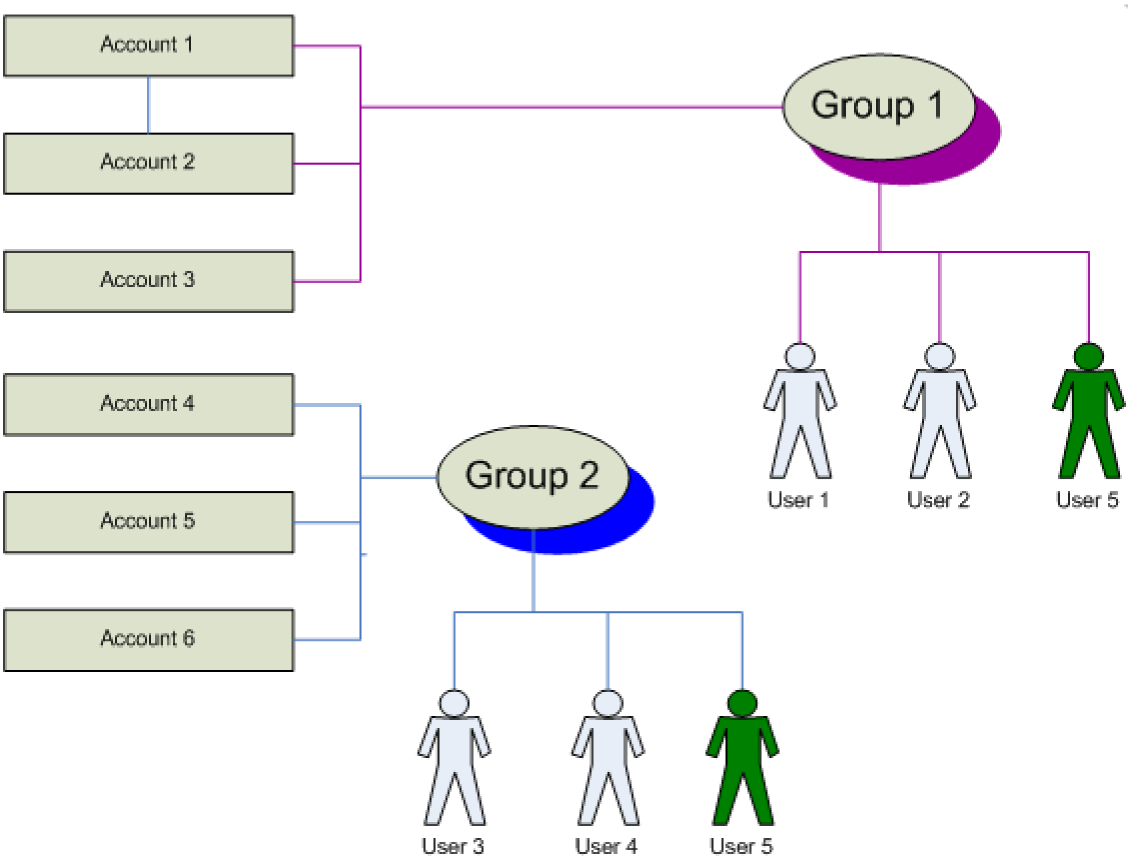
Consider the following examples, each of which shows inverse restriction groups of users and accounts.

Type A Inverse - Usage Example 1



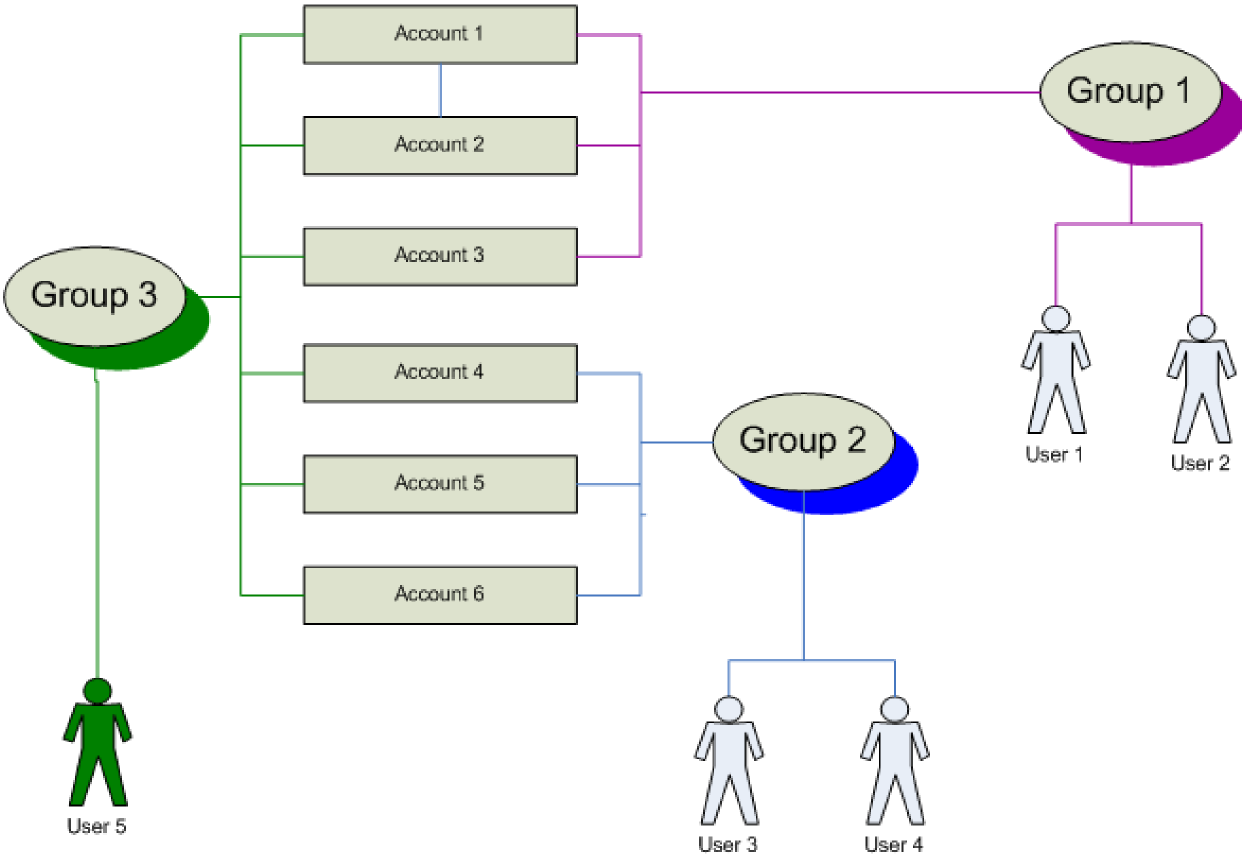
User 1 and User 2 are denied access to Accounts 1, 2, and 3; all other users have access to these accounts. User 3 and User 4 do not have access to Accounts 4, 5, and 6, but other users may access these accounts. (note that User 1 and 2 can access Accounts 4, 5, and 6 though, and similarly User 3 and 4 can access Accounts 1, 2, and 3)

Type A Inverse - Usage Example 2



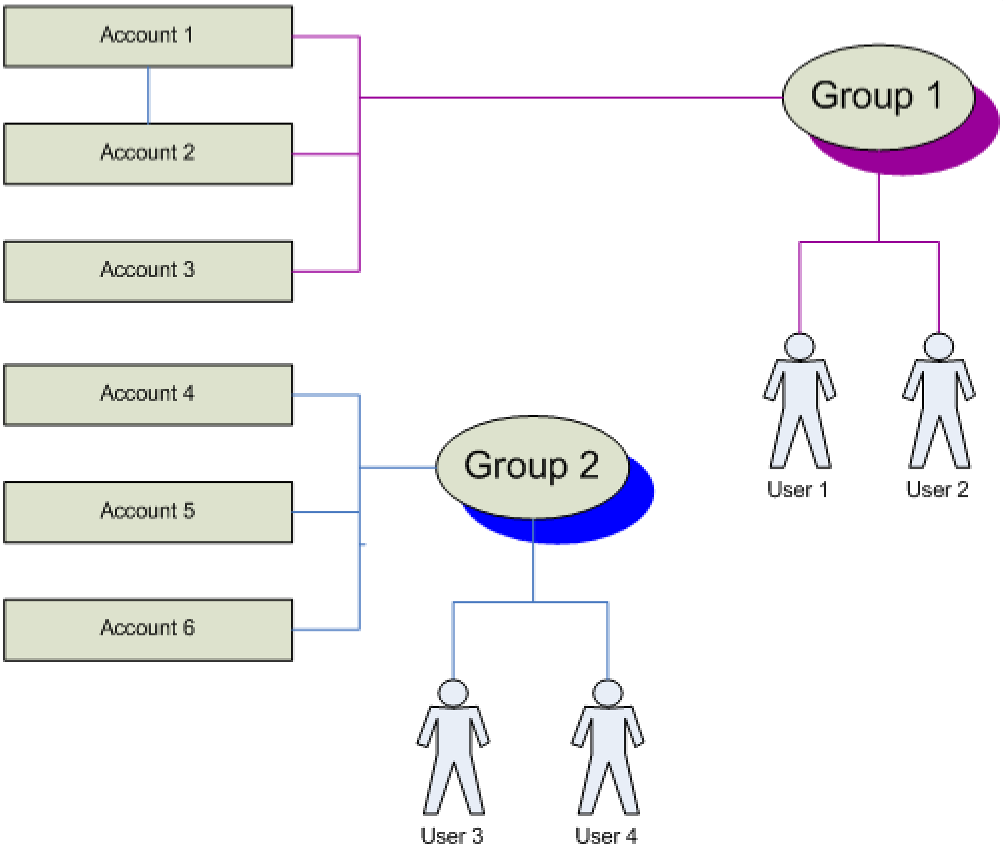
User 5 (who is in both Group 1 and Group 2) does not have access to Accounts 1 through 6. User 1 and User 2 do not see Accounts 1, 2, and 3. User 3 and User 4 do not see Accounts 4, 5, and 6.

Type A Inverse - Usage Example 3



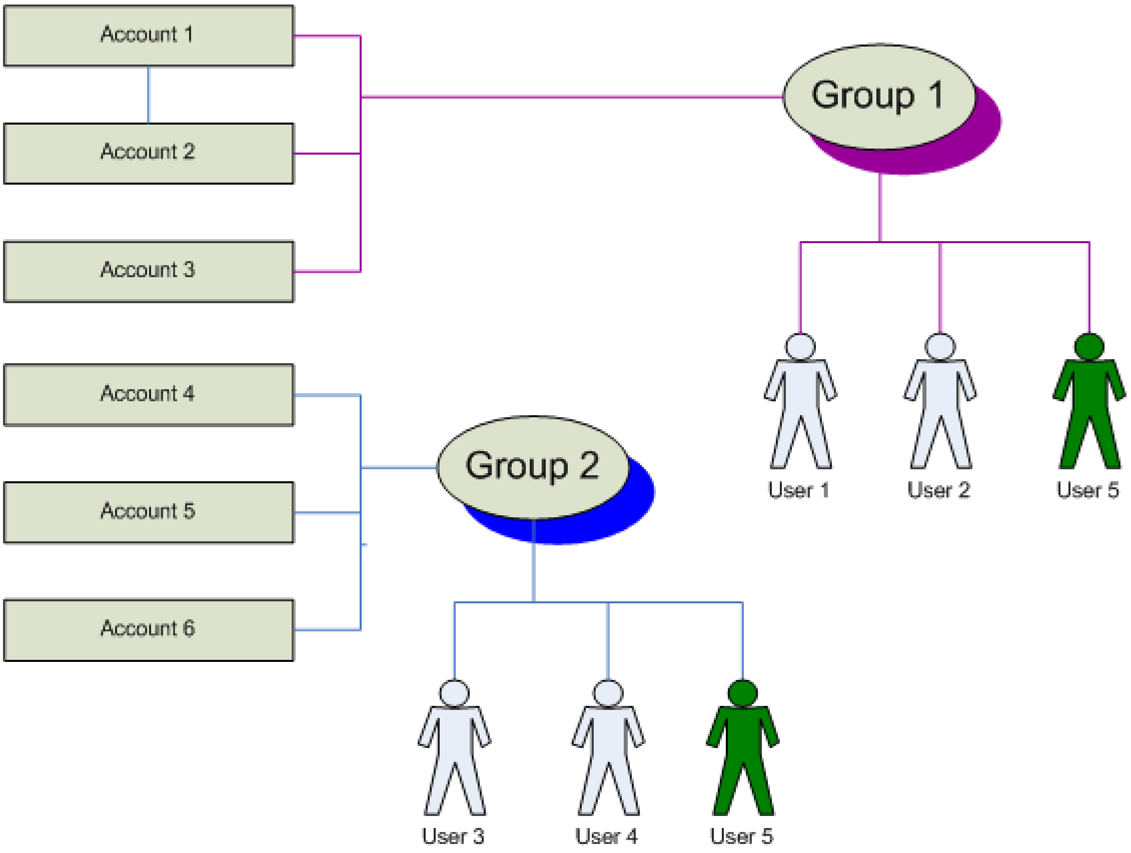
All users have access to all accounts because multiple groups include each of the six accounts.

Type B Inverse - Usage Example 1



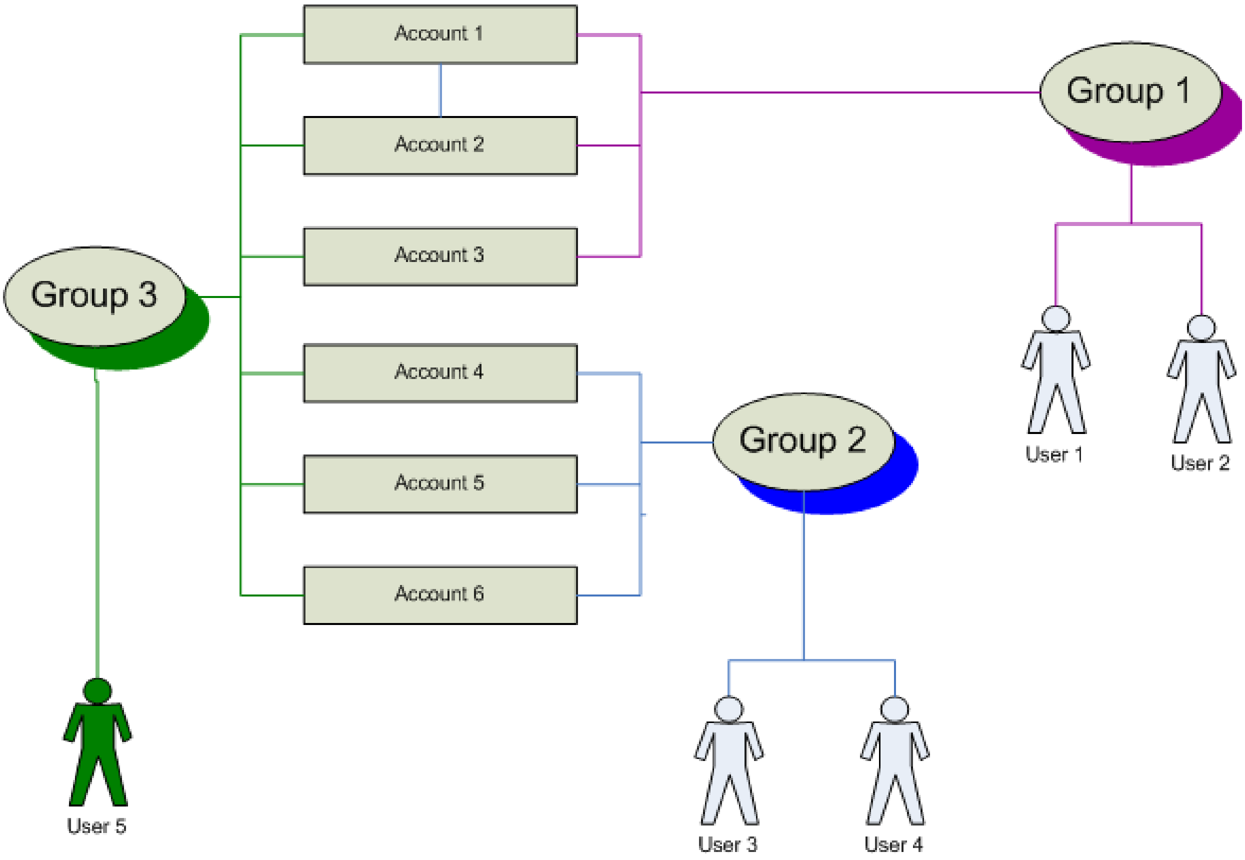
As with the A Inverse groups for this example, User 1 and User 2 are denied access to Accounts 1, 2, and 3; all other users have access to these accounts. User 3 and User 4 do not have access to Accounts 4, 5, and 6, but other users may access these accounts.

Type B Inverse - Usage Example 2



Groups in this example work just as they did for A Inverse groups. User 5 does not have access to Accounts 1 through 6. User 1 and User 2 do not see Accounts 1, 2, and 3, and User 3 and User 4 do not see Accounts 4, 5, and 6.

Type B Inverse - Usage Example 3



Here, the B Inverse groups work differently than A Inverse groups do. User 5 does not have access to Accounts 1 through 6. User 1 and User 2 do not see Accounts 1, 2, and 3. User 3 and User 4 do not see Accounts 4, 5, and 6.

Entities controlled by Restriction Groups

Although the system places no obvious limitations on including entities of various types as members of restriction groups, only groups with entities of specific types actually restrict the use of the entities during run-time because they are supported in application code. Acumatica ERP supports several typical groups with entity types that might be used to suit the security needs of most organizations.

Restriction groups with entities of any types can be created using the Restriction Groups (SM.20.10.30) form. You can also use forms that are usually located in the Setup sections of the applicable modules (mentioned in the lists below) to add restriction groups that apply to entities defined in the modules.

The most commonly used restriction groups in Acumatica ERP are groups that include users as members and grant only particular users access to certain entities in the system. The following types of restriction groups with users as members are supported:

- **Users and GL Accounts**: To control user access to specific accounts
- **Users and Subaccounts**: To control user access to specific subaccounts

- **Users and Subaccount segment values**: To control user access to specific subaccount segment values
- **Users and Vendor Accounts**: To control user access to specific vendor accounts
- **Users and Customer Accounts**: To control user access to specific customer accounts
- **Users and Budget Articles**: To control user access to specific budget articles
- **Users and Warehouses**: To control user access to specific warehouses
- **Users and Inventory Item Classes**: To control user access to specific item classes
- **Users and Inventory Items**: To control user access to specific stock or non-stock items
- **Users and System Email Addresses**: To control user access to specific accounts

Groups without users are used to create special relations between the included entities. Acumatica ERP supports groups with the following types of entities:

- **General Ledger Accounts and Subaccounts**: To define relations between accounts and specific subaccounts. In groups of this specific type, subaccounts are objects of restriction.
- **Branches and General Ledger Accounts**: To define branch-specific accounts.
- **Branches and Subaccounts**: To define which subaccounts should be used by which branches?
- **Inventory Items and Subitem Segment Values**: To define properties of specific inventory items, such as sizes, colors, and materials.

Default Restriction Groups

You may need to restrict user access to numerous similar entities, such as vendors of specific vendor classes, or customers of specific customer classes. Acumatica ERP provides the functionality of default restriction groups assigned to customer and vendor classes.

For instance, you may want to give only a trusted manager access to vendors of one specific class. You can open the Vendor Access (AP.10.20.00) form, create a group, and include the manager in this group. Instead of including every vendor of the class in this restriction group manually, you can save time and include all vendors automatically as follows:

1. Assign this restriction group as the default group for the selected vendor class using the Vendor Classes (AP.20.10.00) form.
2. Click Apply Restriction Settings to All Vendors on the form toolbar to include all vendors of the class in the default restriction group.

To exclude vendors of the class from a restriction group they were members of, on the Vendor Classes form, clear the field for the default restriction group and click Apply Restriction Settings to All Vendors.

To set up a restriction group similarly for customers, follow the instructions above but instead use the Customer Access (AR.10.20.00) and Customer Classes (AR.20.10.00) forms.

Account and Subaccount Security

Acumatica provides restriction groups to enable organizations to control which users can view or use certain General Ledger accounts and subaccounts. Users of the system who hold the same job positions may be related to different branches and generally may not be permitted to view accounts or subaccounts of another branch. Also, within branches, your organization may have sensitive or confidential accounts to which access should be blocked for most users.

When you create a restriction group and include users, accounts, or subaccounts into it, you restrict access to the chosen accounts or subaccounts for all the users except those who are members of the group. Users who are not members of any group will not be able to view the objects included in the groups on data entry forms, financial reports, inquiries, or anywhere in the system.

Membership in restriction groups is checked during runtime on all data entry, inquiry, and processing forms. Membership in groups is not checked for the forms through which users create restriction groups and modify their membership.

The security-oriented forms that relate to the General Ledger module, listed below, allow you to quickly add specific accounts, subaccounts, and even subaccount segment values to any existing restriction group or to create a new group:

- **GL Account Access (GL.10.40.00) form**: Use it to completely configure a restriction group that manages users' access to accounts, subaccounts, and subaccount segments.
- **Restriction Groups by GL Account (GL.10.40.20) form**: Use it to quickly add an account to any existing group. Also, you can remove accounts from any group by clearing their Included check boxes.
- **Restriction Groups by Subaccount (GL.10.40.30)**: Use it to quickly add a subaccount to any existing group. Also, you can remove subaccounts from any group by clearing their Included check boxes.

Restriction Groups for Subitems

To allow you to trace various types of products in the system, Acumatica supports subitems in addition to inventory items. If subitems are used in your system, subitem codes are used with each inventory ID associated with a stock item, even if the product does not have variations to be designated by subitems.

When you enter, for example, a sales order, you should enter an inventory ID and a subitem code for each product. To enter a subitem code, you will need to browse through all possible values of each segment, although most combinations make no sense. Entering the wrong codes can affect item availability data and decrease sales. To avoid these problems, you can use restriction groups.

Example of Using Subitems in Inventory

The merchandise sold by a company is men's apparel: T-shirts, shoes, and socks and Soaps. Suppose we have only three products with the following properties at our warehouse:

Product Type	Color	Size	Material
T-Shirt	Yellow White Red	S, M, L, XL, XXL	Cotton Viscose

Socks	Black	6, 7, 8, 9	Polyester
Shoes	Black Brown White	38, 39, 40, 42, 42, 43	Nubuck Leather
Soap	n/a	n/a	n/a

Further, suppose that we have configured the following segments for the subitem codes:

Segment	Property	Length	Segment Values
1	Color	3	000, YLW, WHT, RED, BLK, BRN
2	Size	3	000, S_;_;, M_;_;, L_;_;, XL_;, XXL , _;_;6, _;_;7, _;_;8, _;_;9, _;38, _;39, _;40, _;42, _;42, _;43
3	Material	9	000000000, Nubuck, Leather, Polyester, Cotton, Viscose

When entering subitem codes on data entry forms, it's easy to enter by mistake a subitem that doesn't apply to any product. For example, in this scenario, for inventory ID SHOES, there is no such subitem as RED-XXL-Polyester, and for inventory ID SOCKS, there is no such subitem as YLW-42-NUBUCK.

To reduce the rate of input errors, you can use restriction groups. The values in each subitem segment are associated with different products, and no values apply to all products. If we create a restriction group that includes an inventory ID and only the segment values that apply to the product under this ID, those entities (inventory ID and specific segment values) will be used only with one another.

Subitem Group	Inventory ID	Segment 2	Segment 3
SHOE SIZES	SHOES	38, 39, 40, 42, 42, 43	
SOCK SIZES	SOCKS	6, 7, 8, 9	
TSHIRT SIZES	TSHIRTS	S, M, L, XL, XXL	
SHOE_MATERIALS	SHOES		Nubuck, Leather
SHIRT_MATERIALS	TSHIRTS		Cotton, Viscose
SOCK_MATERIALS	SOCKS		Polyester

Whether you create groups for colors used for different products depends on the products you handle, but the same segment value cannot be used in two or more groups. Theoretically, if the sets of colors for different products do not intersect and you do not plan to sell these products with other colors, you can create separate groups by colors. But in the example above, a single COLORS group can be created with the following entities:

- **Inventory IDs:** SHOES, TSHIRTS, SOCKS

- **Segment values:** BLK, BRN, YLW, WHT, RED

If all products have colors, there is no need to create a restriction group for the color segment.

As you create an inventory item record, specify in which restriction groups the item is included. For example, when creating a record for a new shoe model, include the inventory ID in the SHOE_MATERIALS and SHOE SIZES groups. Then when you create a sales order for shoes and specify segment values for the subitem, you will see in the second segment only the sizes and in the third segment only materials that apply to shoes.

> As you plan and configure these restriction groups, keep in mind that a segment value can be included in only one restriction group. Also, each segment should have one "not applicable" value (such as zero) to be used with items that do not have this property. Consider an example for soap; as an inventory item, it should be accompanied by a subitem, whose segment values will be chosen as follows:
>
> - Color: 000 (n/a)
> - Size: 00 (n/a)
> - Material: 000000000(n/a)
>
> That is, the subitem for soap will be the following string: 00-000-000000000.

Configuration of Subitem Restriction Groups

You configure subitem restriction groups by using the following steps, some of which occur during early system implementation and some of which happen later:

1. Create subitem restriction groups using the Restriction Groups (SM.20.10.30) form. On the same form, add subitem segment values and inventory IDs to appropriate restriction groups.
2. Specify subitem restriction groups to be used in the inventory module using the Subitem Restriction Groups tab of the Inventory Preferences (IN.10.10.00) form.
3. For each item class on the Item Classes (IN.20.10.00) form, select restriction groups to be used with inventory items of the class.
4. Create inventory items using the Stock Items (IN.20.25.00) form. By default, the item will be included in the subitem restriction groups specified by the item class. If needed, assign the item to other restriction groups if class settings don't match the item properties.

Inventory and Warehouse Security

Because inventory is such a valuable asset to your company, access to inventory records, kit specifications, and prices should be allowed to only a very small number of employees. By limiting access to information about warehouses, inventory items, and transactions, management can properly control stock levels and avoid fraud. By using roles and restriction groups, you can restrict employee access to inventory information based on their positions and job responsibilities. Also, user authorization requirements (username and password) ensure a clear audit trail of changes made to records in the database.

Role-Based Access

With Acumatica's role-based approach to security, you assign users to one or more roles, which are granted various access rights to system objects; changing a role's access rights affects all users assigned to that role. You create and populate roles by using the User Roles (SM.20.10.05) form, and you define access rights for roles on the Access Rights by Role (SM.20.10.25) form. Each warehouse employee is assigned one or more roles in accordance with his or her responsibilities.

For the Inventory Management (IN) module, you can configure roles' access rights to functionality in accordance with employee positions or job descriptions. Consider the following examples of configuring roles' access rights within the IN module:

- While a user with one role may be allowed to accept goods, a user with another role may issue them from a warehouse.
- One role may allow a user with the role to enter count data, and another role may allow a user with the role to initiate or complete count or edit the count data.
- One role may be permitted to prepare replenishment, while another role should be allowed to create purchase orders to replenish the stock.
- Data entry clerks may have a role allowing them access to the data entry forms, while supervisors may have a role giving access to maintenance and processing forms and only higher-level managers may have a role granting access to the Inventory Preferences (IN.10.10.00) form and other security-related forms.

Restriction Group–Based Access

In addition to implementing functional security based roles, you can configure for particular employees different access rights to different entities of the same type: For example, John may have access to wholesale and retail warehouses, while Jane has access to only the retail warehouse record. Acumatica allows you to control user access to specific warehouses and inventory items using restriction groups.

To configure access restriction to warehouses:

1. Create restriction groups to be used to restrict access to warehouses by using the Warehouse Access (IN.10.20.00) form.
2. Assign users to restriction groups.
3. Select a warehouse on the Groups by Warehouse (IN.10.20.10) form, and specify the groups that may have access to this warehouse.

To configure access restriction to inventory items:

1. Create restriction groups to control access to certain specific items.
2. Assign users to restriction groups as follows:
    - By using the Groups by Item Class (IN.10.30.10) form, select an item class and specify which of the groups will have access to items of the class by default. The groups

selected for the item class will appear on the Subitem/Restriction Groups tab of the Item Classes (IN.20.10.00) form.

- On the Groups by Inventory Item (IN.10.30.20) form, select an inventory item and specify which of the groups will have access to the item.

Questions

1. What are restriction groups and how they differ from roles?
2. What are the various types of restriction groups available in Acumatica ERP and define each in brief.
3. List various entities that are controlled by restriction groups. Discuss practical usability/scenario of each one.

Hands on – Row-Level Security

The exercises below will help you become familiar with using various features in row-level security.

Scenario 1: Customer Account Restrictions

Create restriction groups such that following users does not have access to respective customer accounts. All other users should have access to all the customer accounts.

Users	**Customer Accounts**
Chubb, Church, Cozzi, DRick	ABCSTUDIOS
Andrews, Baker	AMROBANK, CRABTREE
Becher, Bernia, Bloom	GOLDENKEY, GOLDRIVER
Lai, Norman	LASERWORKS

Scenario 2: Vendor Account Restrictions

Create restriction groups such that following vendors are available only to these users. All other users should not have access to these vendors.

Vendor Accounts	Users
SONICSOFT, DEWSOFT, DBCONSULT	Chubb, Church, Cozzi, DRick

Scenario 3: Inventory Item Restrictions

Create restriction groups such that following items are not available to these users. All other users should have access to all the items.

Inventory Items	Users
ACCOMODATION, DESIGN, OVRSEATRAVEL	Chubb, Church, Cozzi, DRick
CPU00001, CPU00002, CPU00003	Andrews, Baker, Church

Scenario 4: GL Account Restrictions

Create restriction groups such that following accounts are available to these users. All other users should not have access to these accounts.

GL Accounts	Users
100000, 101000, 101010, 101020	Chubb, Church, Cozzi, DRick
200000, 200010, 200020	Andrews, Baker, DRick

Row-Level Security Form Reference

GL Account Access

You can use this form to create and modify restriction groups for managing user access to accounts and subaccounts.

GL Accounts by Branch Access

You can use this form to create restriction groups to specify which of the branches have access to specific General Ledger accounts.

Subaccounts by Branch Access

You can use this form to create restriction groups and to specify which of branches have access to which of subaccounts.

GL Budget Access

You can use the GL Budget Access form to create restriction groups for managing user access to budget articles, as well as to modify existing restriction groups by adding or removing particular budget articles or users.

Vendor Access

Use this form to create restriction groups for managing user access to vendor accounts, or to modify existing restriction groups by adding or removing specific vendor accounts or users. Also, restriction groups can be created on the Restriction Groups (SM.20.10.30) form.

Customer Access

By using this form, you can create restriction groups for managing users' access to customer accounts. You can also use the form to modify existing restriction groups by adding or removing specific customers or users. Restriction groups can also be created using the Restriction Groups (SM.20.10.30) form.

Warehouse Access

By using this form, you can create restriction groups for managing user access to warehouses, or modify these groups. You can use the form to modify the list of warehouses to which users of the specific group have access. The Restriction Groups by Warehouse (IN.10.20.10) form can also be used to manage user access to warehouses.

Inventory Item Access

By using this form, you can create restriction groups for managing user access to stock items. You can also modify existing restriction groups by adding or removing particular users, stock items, and item classes. The Restriction Groups by Item Class (IN.30.30.10) and Restriction Groups by Item (IN.10.30.20) forms can also be used to manage user access to stock items and classes.

Email Account Access

On the Email Account Access form, you can create restriction groups for managing user access to the system email accounts. It also enables you to modify existing restriction groups by adding or removing users who may access the specific email accounts.

Restriction Groups by User

You can use the Restriction Groups by User form to assign a user to a restriction group (or multiple groups) or to remove the user from restriction groups.

Restriction Groups by GL Account

You use this form to select an account and view and edit its membership in the available restriction groups.

Restriction Groups by Subaccount

You can use this form to select a subaccount and view and edit its membership in the available restriction groups.

Restriction Groups by Sub Segment

You can use this form to select a subaccount segment value and view and edit its membership in the available restriction groups.

Restriction Groups by Branch

By using this form, you can select a branch and view and edit its membership in the available restriction groups.

Restriction Groups by Budget Article

By using this form, you can select a budget article and view its membership in the available restriction groups.

Restriction Groups by Customer

By using this form, you can select a customer account and view the restriction groups it belongs to. You can quickly add the customer account to a group or exclude it from a group.

Restriction Groups by Vendor

Use this form to select a vendor account and view the restriction groups it belongs to. You can quickly add the vendor account to a group or exclude it from a group.

Restriction Groups by Warehouse

Use this form to select a warehouse and view or change the restriction groups that include it.

Restriction Groups by Item Class

Use this form to select an item class and view or change the restriction groups that have access to this class.

Restriction Groups by Item

By using this form, you can select a stock item and view or change the restriction groups that have access to it.

Restriction Groups

You use restriction groups, which you create or modify by using this form, to implement security goals that cannot be achieved through roles.

Restricted Entities

You use this form to modify a restriction group by including or excluding its members (which are system entities, including users). If a group includes particular users, only those users have access to the group entities. If a group doesn't include users, the entities of different types included into the group can be used only with group members.

Branch Security Administration

Learning Objects:

- *Learn how to secure Branches in Acumatica ERP*

If your organization has multiple branches, you may need to control which employees get access to which branches. Because branches share some data, you may also need to control access to the shared data. Acumatica ERP provides various options for branch security administration, which are outlined below.

Branch Access Role

During Acumatica ERP implementation, your administrators configure branches to match your organization's structure and reporting needs. By default, at least one branch is required as the organization itself. If your organization has only one branch, there is no need to control access to the data by branch.

If your organization has more than one branch, you may configure user access to the branch-specific data: For each branch, you can create a role, include all the branch employees as the role's users, and assign the role to the branch.

If no role is assigned to any of branches, all users have access to all the data. Once a role is assigned to one of the branches, roles will be required to access other branches, and a branch with no role assigned will be inaccessible to any user.

Because only users with a particular branch access role can access the branch data, to allow particular users to access multiple branches, assign them all the required roles.

Access Rights Provided by a Branch Access Role

If a user, by his or her functional role, has access to a data entry form where this user enters a document and specifies the branch of origin, only branches to which the user has access are available on the drop-down list. The users who have access to multiple branches, can choose the specific branch from the Branches menu on the form toolbar and create documents on behalf of the chosen branch.

A user may, by his or her functional role, have access to the forms used to configure inter-branch functionality—such as Branch Account Mapping (GL.10.10.10), Groups by Branch (GL.10.30.20), Branches (CS.10.20.00), Buildings (CS.20.50.10), and forms related to restriction groups. Such a user can view all the branches, regardless of the branch access role the user is assigned to.

Certain users, by their roles, are authorized to create assignment maps and workgroup hierarchies and have access to the Assignment Rules (EP.20.50.00), and Company Tree (EP.20.40.60) forms. These users will have the same level of access (as suggested by their roles) no matter which branch they belong to.

Restriction Groups

Branches may have some data shared between branches and some data kept as branch-specific. As such, you may need to restrict user access to data that is functionally shared but may contain sensitive data, such as some General Ledger accounts and subaccounts. Acumatica ERP provides restriction groups to allow you to control which accounts and subaccounts are used with which branch.

To use this functionality, by using appropriate forms in the Configuration section of the General Ledger module, create a couple of restriction groups for each branch. Include in one such group the branch and all accounts that should be used only by this branch and include in another group the branch and subaccounts to be used only by this branch. We do not recommend that you include both accounts and subaccounts in a group designed for a branch. Restrictions on subaccounts and accounts to be used by a specific branch are implemented through separate types of groups: branch-accounts groups and branch-subaccounts groups. Including accounts and subaccounts in one group with a branch would apply additional restrictions that you probably don't want: The listed accounts could not be used with subaccounts other than those included in the group.

We also do not recommend that you assign users to any groups with branches as group members. User access to branches should be configured by using access roles.

Questions

1. Why are branch restrictions important and when they should be used?

2. Branch restriction scenario 2
   Following users should be restricted to use only these branches and accounts. These accounts should not be accessible to other users.

Users	Branches	GL Accounts
Chubb, Church	EAST	100000, 101000
Cozzi, DRick	NORTH	101010, 101020

Hands on – Creating Branch Restrictions

This exercise will create users that are restricted to specific branches.

Verify System Settings

1. Open ***Enable/Disable Features*** screen (CS.10.00.00)

    a. Configuration > Common Settings > Licensing > Enable/Disable Features

2. Verify that ***Finance > Branch Accounting*** is checked

    a. If not checked, click the ***Modify*** button at the top of the screen, check the box, and then press the ***Activate*** button.

Create a Branch Roles

3. Open ***User Roles*** screen (SM.20.10.05)

    b. Configuration > User Security > Manage > User Roles

4. Create the East Users role by adding a new record with the following values:

Field	**Value**
Role Name	EAST Users
Role Description	EAST Users
Guest Role	{blank}
Membership	{blank}

5. **Save** your changes

6. Repeat steps 3-5 for the North, West, and South Users.

Configure Branch Access Roles

7. Navigate to ***Branches*** screen

    a. Organization Structure > Configure > Branches

8. Set the **East Branch** role by setting the information below:

Form or Tab	Field	Value
Summary Section	Branch ID	EAST
General Info	Access Role	EAST Users
All others	All fields	{keep existing settings}
Summary Section	All other fields	{accept default}

9. **Save** your changes.

10. Repeat steps 7-9 for the North, West, and South branches.

Create Users and Add to Branches

1. Navigate to ***Users*** (SM.20.10.10) screen

    a. *Configuration > User Security > Manage > Users*

2. Create User with the information below:

Form or Tab	Field	Value
Summary Section	Username	EastUser
Summary Section	Generate Password	{unchecked}
Summary Section	Password	123
Summary Section	First Name	East
Summary Section	Last Name	User
Summary Section	Email	{any valid email address}
Summary Section	All other fields	{accept default values}
Roles	Role Name -> Administrator	{checked}
Roles	Role Name -> EAST User	{checked}
Roles	Role Name -> Internal User	{checked}
Roles	All Other Roles	{unchecked}
Statistics	All Fields	{accept default}
IP Filter	All Fields	{accept default}

3. Optional: create similar users for other branches.

4. Now that we have created the additional roles which restrict access to branches, we need to add access to those branches to our administrator account.

    a. On users screen, select the administrator and check boxes next to all the new branch roles that you assigned to specific branches.

Login and Verify

1. Logout as administrator and login using the user account that you just created

    a. *Username: EastUser*

   b. *Password: 123*

2. Things to notice

   a. The Branch selector at the top of the screen only includes the East Branch.

   b. If you navigate to the ***Journal Transactions*** (GL.30.10.00) screen, there are no transactions to view because all existing transactions belong to the main branch.

   c. If you navigate to the Vendors (AP.30.30.00) screen, you will see the list of vendors the same as you did before. But, in the vendor summary inquiry (AP.40.10.00), you will not see any values because the transactions all occurred in the Main Branch.

   d. Explore other areas

3. Login as administrator and verify that you have full access as you did before.