QAD Enterprise Applications
Enterprise Edition

User Guide
QAD Security and Controls

78-0841A
QAD Enterprise Applications 2010
Enterprise Edition
March 2010

Security_and_Controls_UG_v2010EE.pdf/dmk/dmk

QAD Inc.
100 Innovation Place
Santa Barbara, California 93108
Phone (805) 566-6000
http://www.qad.com

Contents

Chapter 1

Introduction

This section introduces the security and internal control features in your system.

Overview

The security and related internal controls operating in your system must be viewed within the context of your organization's overall security framework. While it is beyond the scope of this user guide to discuss the details of information security, the fundamental components involve measures to assure the preservation of:

- Confidentiality—ensuring that information is accessible only to those authorized to have access
- Integrity—safeguarding the accuracy and completeness of information and processing methods
- Availability—ensuring that authorized users have access to information and associated assets when required

Security properly starts with a comprehensive policy statement that:

- Demonstrates clearly management's support and commitment to security
- Defines the principal security components important to the organization
- Describes the general approach for meeting security objectives

After the policy statement is prepared, procedures, guidelines, and other supporting administrative controls are typically defined to support the policy. Finally, technical controls are designed and implemented to support the administrative controls.

The system provides multiple types and levels of security and internal controls, which are described in this chapter. This chapter also includes several checklists to use as starting points in planning and implementing a comprehensive security plan to meet the specific security requirements of your environment. See "Security Planning Checklists" on page 6 for details.

The specific level of security control an organization should implement is a function of the underlying information security requirements. Those requirements originate:

- Externally, including regulatory, legal, and legislative requirements
- Internally, based on the value of information assets, associated risks to those assets, and available controls that can eliminate or mitigate exposures to an acceptable level

Much of the security control in the system is designed to support external requirements. Numerous controls to support customers who are concerned with meeting the security requirements of legislation and regulations such as the Sarbanes-Oxley Act and Food and Drug Administration 21 CFR Part 11.

Security

The security model used by the system integrates the different components of the system architecture, controls who can access the system, and defines the actions that system users can perform.

Using security features, you can configure system login behavior, define password strategies, create and maintain users and roles, as well as specify user access to domains and entities.

The guiding rule in role-based security is that access to a resource is not allowed unless it has been specifically granted. Role-based security features let you control user access to all menu-based application resources, as well as some resources that represent activities that are not directly accessed from the menu.

Menu-level resources are one of two types:

- Standard programs, which display on the system menu as a single maintenance function. Standard programs are available in both the .NET User Interface ((UI) and character interface.
- Component-based functions, which display on the system menu as items with one or more associated activities. A component-based function is always associated with an activity or multiple activities. Component-based functions are available only in the .NET UI.

Using the login security features, you can secure your system from unauthorized users, as well as optionally implement single sign-on to improve ease of access for system users.

You also can configure additional types of security that provide enhanced protection for individual database records, fields, sites, GL accounts, and so on.

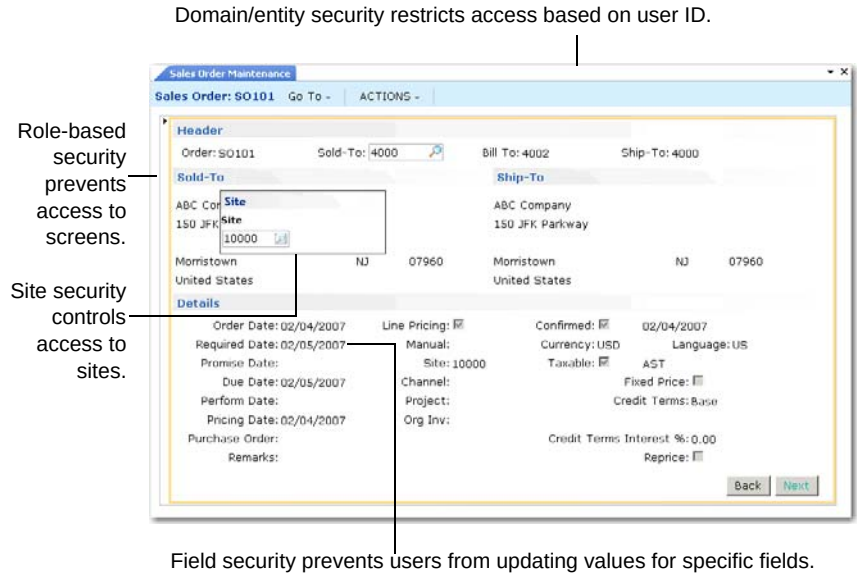
Note If you intend to use other components of the QAD Enterprise Application Suite that communicate with core functions through APIs—for example, QAD QXtend Outbound (QXO)—a system administrator must configure security for these add-on products appropriately. These security details are included in the relevant product documentation of the other components.

When a user logs in, the system determines the programs or functions to display on the application menu based on the user's roles in the current domain and entity. This occurs in exactly the same way regardless of whether login is from the character user interface or the .NET UI.

Important The various system security controls are primarily effective within an application session. The system database should be protected from any unauthorized access, not just access from within an application session. Additional controls should be considered to prevent compromise of system data using other means. See "Operating System and Progress Security" on page 19 for details.

During an application session, several different types of security operate at the same time.

Fig. 1.1
Types of Security



Internal Controls

In addition to extensive security features, the system also has internal control features. Internal controls are mechanisms that help an organization comply with legal or regulatory requirements to reduce their exposure to potential liability imposed for violations. For example, the Sarbanes-Oxley Act of 2002 mandated that public companies must provide an assessment of the effectiveness of the organization's internal control over financial reporting.

The system has these internal control features:

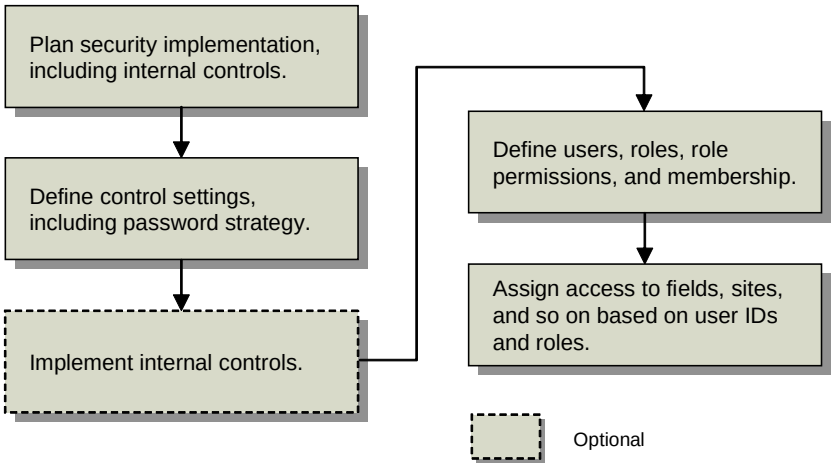
- Electronic signatures. Provides features that require users of some system programs to enter a valid user ID and password before they can create or update records. See Chapter 6, "Using Electronic Signatures," on page 81.
- Auditing. The Auditing module integrates with the Progress OpenEdge Auditing capability in Progress OpenEdge 10. You can configure your system to maintain audit trails. Audit-trail records are created and stored in audit trail tables. They contain facts about changes made in the databases. A typical audit record includes information that helps you identify who made a change, which program made the change, when the change was made, and what the change was. You can set up these functions for all tables or you can limit the audit trail recording activity to specific tables. See Chapter 7, "Auditing," on page 111.
- Segregation of duties. Provides features that prevent a user from participating in more than two parts of a transaction or process. This is accomplished by partitioning the system application resources into mutually exclusive categories.

Note Segregation of duties (SOD) is not included in this release.

Implementation Summary

Figure 1.2 illustrates a workflow for implementing security and internal control features.

Fig. 1.2
Security Workflow



Establishing a Security Plan

Every user must be identified in the system, given access to a domain and at least one entity in the domain, and associated with at least one role in the domain in order to gain system access.

A number of roles are supplied with the system. These roles can be used for notification when a new customer, supplier, employee, or end user is created. These roles are provided to enable system setup; for details see the section “System-Supplied Roles” on page 49.

Use the set of checklists provided in this section as a starting point for determining the focal points to consider when establishing a security plan. See Table 1.1 on page 6.

You should consider both internal and external requirements when planning such security elements as password protection. For example:

- Does your organization have specific internal controls-related requirements that may require the implementation of segregation of duties or update restrictions?

  Important By carefully planning how you will integrate your defined SOD policy with your setup of user roles, role permissions, and role membership, you may avoid SOD policy violations that require configuration rework.
- Does your organization have specific requirements regarding password aging for all its systems?
- Do external regulatory agencies set standards for password complexity, or whether the logged-in user ID should always display on the screen?
- Does your environment require database or operating system security controls implemented outside your QAD applications?

Other planning considerations apply if you are setting up security for a multiple-domain database.

For example, user profiles defined in User Maintenance (36.3.1) apply to all domains in the system. However, profiles include several generalized codes that are domain specific such as access location and user type. To prevent validation errors, you should ensure that these codes exist in all domains.

If you determine how you will use such system-wide data as part of your security planning effort, you can prevent duplication of effort by having basic information in place when you create new domains. Refer to Chapter 3, “Setting Up Financial Foundations,” in *User Guide: QAD Financials.*

Implementing Your Security Plan

After planning how your security system should operate to meet your organization’s specific requirements, perform the following tasks to implement the plan:

- Define control settings using Security Control (36.3.24). An important feature of this program is the Passwords frame, where you establish a system-wide password strategy. See page 25.
- Set up users, roles, role permissions, and role membership. Depending on your overall security plan, you can define such elements as domain access and role membership, as well as enter temporary passwords for your users. See page 35.
- Set up internal controls (optional). You can reduce the complexity of implementing segregation of duties by partitioning your system resources at the same time as you define users and roles by using an iterative approach.

  Note The Segregation of Duties module is not available in this release.
- Set up user access to fields, sites, GL accounts, and inventory movement codes as required. See page 63 for details.

Security Planning Checklists

Tables 1.1 through 1.3 summarize the various security controls that should be considered as part of an effective overall information security plan for the system. The degree to which each of these items is relevant will be a function of an organization’s security requirements.

Where applicable, the tables include references to information on related topics.

Table 1.1
Planning, Policies, and Procedures Checklist

Topic	Reference
Review all information about security documentation for both the system and Progress prior to installation (or software upgrade if applicable).	• This chapter • *Installation Guide* • Progress documents, including *Data Administration, Guide*, *Client Deployment Guide*, and *Programming Handbook*
Review all application-related files to determine the appropriate permission and ownership settings.	“Operating System and Progress Security” on page 19

Topic	Reference
Optionally, determine and document any segregation of duties policy requirement. Also document how application resources should be partitioned.	Setting Up Segregation of Duties
Document the users who should be permitted access to the system and the domains and entities to which they should have access. This step will require an in-depth knowledge of your organization's security requirements.	• "Defining Users" on page 38 • "Specifying Access to Domains and Entities" on page 45
Document the role or roles to which users should be assigned. This step will require an in-depth knowledge of your organization's security requirements.	"Setting Up Roles" on page 47
Determine if specific users will require access to individual fields, sites, general ledger accounts, or inventory movement codes for standard application programs.	"Additional Security for Standard Programs" on page 63
Consider requirements for policies and/or procedures regarding the deactivation of old user accounts. To meet the requirements of many regulated environments, user accounts can be disabled, but not deleted, once they have been used to access the system.	"Setting Up Users" on page 37
Define policies and procedures to be used to assure that user/role information will be kept current.	
Determine procedures to be used to create new user accounts and communicate initial passwords (e-mail, personal contact, other).	"Creating a Password Strategy" on page 30
Decide if a simplified access approach is sufficient. This lets users log in based on operating system-level security.	"OS-Based Login Security" on page 18
Determine if single sign-on will be enabled for users employing the .NET User Interface.	"Single Sign-On Enabled" on page 29
Define how often users are required to changed passwords, and update the corresponding system security setting.	"Expiration Days" on page 31
Define procedures for failed login attempts, including: • The number of failed attempts before an event notification should be communicated to the defined security administrators • Alternatives to e-mail notification • Reviews of system logs • Procedures for resetting locked accounts	• "Setting Up Security Control" on page 25 • "Monitoring System Security" on page 32
Define password policies and procedures, including password composition, length, expiration, and reuse of previous passwords.	"Creating a Password Strategy" on page 30
Define appropriate policies and procedures for users requiring that application sessions be locked using a screen saver or comparable mechanism whenever the user leaves the session unattended.	"Workstation-Level Security" on page 21

Table 1.2
Progress and Operating System Checklist

Topic	Reference
Determine whether to implement Progress as well as user ID and password controls for your system.	"Progress-Level Database Access" on page 21
Determine requirements for Progress-level schema security to control access to application database tables.	"Progress-Level Database Schema Controls" on page 20
Consider disallowing Progress-level table and field access for the blank user ID	"Progress Editor Access" on page 20
Determine the period of inactivity after which a system session should be terminated. For each device used to access the system, ensure that a screen saver or comparable utility is set to activate after the defined period of activity, requiring reentry of the user's password to unlock the application session.	"Workstation-Level Security" on page 21
Determine whether multiple users share a common workstation to access the system and whether appropriate operating system functionality exists to adequately support security.	Operating system documentation

Table 1.3
System Security Parameters, Setup, and Processes Checklist

Topic	Reference
Verify and update relevant system control program settings, especially those for security.	"Setting Up Security Control" on page 25
Define users assigned to the security administrator role, who will receive e-mail notification of security events such as failed logins exceeding a defined threshold.	• "Administrator Role" on page 28 • "Maximum Access Failures" on page 28
Update system security settings regarding user IDs and passwords, including: • Password composition • Password length • Password expiration • Limits on re-use of previous passwords • Limits on number of failed login attempts	"Creating a Password Strategy" on page 30
Determine how system security should be implemented to protect the integrity of database records. For each site, GL account, and so on, specify the appropriate users authorized to access data.	"Additional Security for Standard Programs" on page 63
Review users and roles for potential segregation of duty issues and adjust assignments as appropriate.	Setting Up Segregation of Duties

Security and Internal Controls Programs

Table 1.4 lists the menu programs you use to define and maintain security and internal controls in your system. The system uses a combination of Progress-based (`.p`) and component-based functions that have the form Component.Activity, such as BRole.Create, which is the create activity of the role component.

Table 1.4
System Security Menu (36.3)

Program No.	Description	Program Name
36.3	System Security Menu	
36.3.1	User Maintenance	mgurmt.p
36.3.2	User Inquiry	mguriq.p
36.3.3	User Password Maintenance	mgurmtp.p
36.3.4	User Domain/Entity Access Maintain	BUser.Modify
36.3.5	User Domain/Entity Access View	BUser.UsrCompanyDomainAccess
36.3.6	Role Maintenance	
36.3.6.1	Role Create	BRole.Create
36.3.6.2	Role Modify	BRole.Modify
36.3.6.3	Role View	BRole.View
36.3.6.4	Role Delete	BRole.Delete
36.3.6.5	Role Permission Maintain	BRole.Permissions
36.3.6.6	Role Membership Maintain	BUserRole.Maintain
36.3.6.8	Role Permissions View	BRole.RoleDefinition
36.3.6.9	Role Membership View	BUserRole.RoleMembership
36.3.6.10	User Access View	BUser.Useraccess
36.3.6.11	Role Export	BRole.Export
36.3.6.12	Role Import	BRole.Import
36.3.7	Update Restrictions Menu	
36.3.7.1	Inv Transfer Restriction Maint	mguritmt.p
36.3.7.2	Inv Detail Restriction Maint	mguridmt.p
36.3.7.3	Unplanned Iss/Rct Restrict Maint	mgurirmt.p
36.3.7.5	PO Restriction Maintenance	mgurpomt.p
36.3.7.6	PO Receipts Restriction Maint	mgurprmt.p
36.3.7.8	SO Restriction Maintenance	mgursomt.p
36.3.7.9	SO Shipments Restriction Maint	mgurssmt.p
36.3.7.13	DO Restriction Maintenance	mgurdomt.p
36.3.7.14	DO Shipments Restriction Maint	mgurdsmt.p
36.3.7.15	DO Receipts Restriction Maint	mgurdrmt.p
36.3.7.17	SSM Restriction Maintenance	mgursmmt.p
36.3.7.19	Update Restriction Report	mgurrp.p
36.3.13	Operational Security Menu	
36.3.13.1	GL Account Security Maintenance	mgacsmt.p
36.3.13.2	GL Account Security Report	mgacsrp.p
36.3.13.8	Site Security Maintenance	clsismt.p
36.3.13.9	Site Security Report	clsisrp.p
36.3.13.13	Inventory Movement Code Security	sosimt.p
36.3.13.14	Inv Mvmt Code Security Browse	gpbr502.p
36.3.15	Field Security Menu	
36.3.15.1	Field Security Maintenance	mgflpwmt.p

Program No.	Description	Program Name
36.3.15.2	Field Security by Role	mgflgpmt.p
36.3.15.3	Activated Field Security Report	mgflpwrp.p
36.3.15.4	Dictionary Field Security Report	mgfldcrp.p
36.3.15.6	Component Field Security Create	BFieldSecurity.Create
36.3.23	Reports and Utilities Menu	
36.3.23.1	Logon Attempt Report	mgurpsrp.p
36.3.23.2	User Account Status Report	mguactrp.p
36.3.22	User Access by Application Inquiry	lvusriq.p
36.3.23.12	User Password Force Change Util	utfrcpsw.p
36.3.24	Security Control	mgurpmmt.p

Chapter 2

Security Overview

This section discusses the security features available in your system:

Role-Based Access Security

Role-based access security is a mechanism that governs a user's ability to gain access to domains, entities, and menu-level functions.

Role-based access security operates through the use of several key components:

- Roles
- Role permissions
- Role membership

The system has additional types of security that can be configured for standard programs and component-based functions.

Roles

A role is a logical subset of activities that describes a user's business function or set of responsibilities within a business enterprise. You can define as many roles as required in the system in order to model your business processes. Roles are created by using Role Create (36.3.6.1).

Typically, a user in the system has at least one role—and possibly several roles. In addition, the same role can be associated with several users. Before users can log in to the system, they must be associated with at least one role.

A role, when associated with a set of application resources, defines the tasks or activities a user can perform when using the system. The process of associating application resources to a role defines role permissions. For details see "Role Permissions" on page 12.

Roles operate within the context of the domains and entities to which the user has been granted access. This concept is known as role membership. For details see "Role Membership" on page 13. A user with multiple roles has access to the sum of the resources assigned to each.

Role Permissions

Role permissions are defined by assigning a set of application resources to a role using Role Permissions Maintain (36.3.6.5).

- For component-based functions, role permissions control the ability to use the various types of activity—create, modify, view, and delete, for example.
- For standard programs, role permissions control the ability to execute those programs.

Note Access control can also be defined for fields, sites, GL account updates, and inventory movement codes using user ID, role, or a combination. For details see "Additional Security for Standard Programs" on page 63.

The application resources defined in the system display in a tree layout similar to the way the menu looks in the .NET User Interface. To define role permissions, you select the resources to assign to the role. Once role permissions and role membership have been defined, when a user opens a workspace, only the application resources associated with that user role display on the application menu. When a user has more than one relevant role, the application resources that display are essentially the sum of the user roles.

Example Sophie Woods has been assigned the roles Project Manager and Accountant. The Project Manager role allows her access to the Customer View function. The Accountant role allows her access to the Customer Invoice Create function. Consequently, the following menu choices display when she logs in:

Customer View

Customer Invoice Create

The role-based security that is defined for a function also applies to any associated functions that are available on the Go To menu for which a user has been granted access. For example, if you are modifying data in the Customer Invoice Create function, the Go To menu for that function displays related functions—Daybook Create, for example—for which you have appropriate permissions.

Role-based security also applies to parts of the system that do not have a user interface—for example, Web services and API calls, as well as daemons. For more information on configuring daemons, see *User Guide: QAD System Administration*.

Role Membership

Role membership associates users and roles, as well as the domains and entities in which that role operates. Use Role Membership Maintain (36.3.6.6) to create and maintain role memberships.

For each domain, access can be restricted to one or more entities in the domain. In essence, role membership defines the *context* of a particular role by specifying the meaning of a role within a specific domain and entity.

A user's role always operates within the context of a domain and entity; you cannot set access at the domain level. You must explicitly grant access to users to each entity within the domain. However, entity-level access has meaning in most cases only within financial functions. Users who will be working exclusively with operational functions such as sales, shipping, and manufacturing are typically given access to the primary entity of the domain.

Example Sophie Woods has the role Project Manager for all companies in the Australia domain. When she accesses the Australia domain, the access privileges for her Project Manager role apply for all entities within the domain. Her privileges do not apply if she logs in to a different domain.

Example Roger Spencer has been assigned the role Accountant, but only for the entity 001 Fit & Co Pacific in the Australia domain; his role privileges do not apply for other entities in the Australia domain or any other domain.

Certain standard programs, described in the next section employ a user ID, role, or sometimes both in order to control access, as in previous versions of QAD applications.

Additional Types of Security

Some additional types of security can be configured for standard programs and component-based activities.

Additional Security for Standard Programs

The system has several types of security that apply to operational programs only. In these programs, security is defined by user ID, role, or a combination of both.

- Field Security Maintenance (36.3.15.1) limits who can update specific fields. For field security, specify a user ID.
- Update restrictions functions on the Update Restrictions menu (36.3.7) limit who can update specific records and create specific issue, receipt, and transfer transactions.
- GL Account Security Maintenance (36.3.13.1) restricts access to GL accounts from operational functions. Specify any combination of user IDs or roles.
- Site Security Maintenance (36.3.13.8) limits who can create inventory transactions at secured sites. Specify any combination of user IDs or roles.
- Inventory Movement Code Security (36.3.13.13) lets you grant or deny user access to shippers and other transactions using specific movement codes at a site. Specify any combination of user IDs or roles.

For details about setting up operational programs, see “Additional Security for Standard Programs” on page 63.

Additional Security for Component-Based Functions

You can also set up field security for component-based functions. This type of field security is more complex and detailed than field security for standard programs. You define by role whether fields are enabled, disabled, or hidden.

To use field security, you must first enable the system-wide setting using System Maintain (36.24.3.1).

For details on component-based field security, see “Additional Security for Component-Based Functions” on page 60.

Password Management

The system offers a flexible approach to assigning and managing passwords, based on the specific requirements of each environment.

Settings in Security Control (36.3.24) determine how passwords are generated, structured, and controlled. Your strategy can be as complex or as simple as needed to meet requirements.

You can specify:

- The minimum length of the password, including minimum numbers of numeric and non-numeric characters
- The number of days passwords are valid and whether the system begins warning users of the expiration date a given number of days in advance
- The number of days or password change cycles that must pass before a user can reuse the same password
- The manual or automatic method used to generate temporary passwords

For details, see “Creating a Password Strategy” on page 30.

Example In a high-security environment, you might specify an eight-character password that must contain at least three numbers. Users must change passwords every 60 days, and are warned each time they log in within 10 days of expiration. To prevent even the system administrator from knowing individual passwords, the system is set up to automatically generate new temporary passwords and e-mail them directly to each user. Users must then create their own passwords at the first login using the temporary password—subject to the parameters defined in Security Control.

In case of forgotten or compromised passwords, User Maintenance (36.3.1) lets system administrators force an individual user to change the password at next login. Force Password Change Utility (36.3.23.12) makes all users or specified roles change their passwords. For details, see “Updating Passwords” on page 43.

Login Security

The following types of security are enforced at login:

- Login security determines whether a user can log in to an application session based on their user ID and password. This level of security is always active, although how it is implemented depends on settings in Security Control.

  For example, system administrators can choose to allow valid users to log in to the QAD application based on operating system-level access. See “OS-Based Login Security” on page 18 for details.

  Note You also should consider additional access security options at the operating-system and Progress levels. See “Operating System and Progress Security” on page 19 for details.
- Domain/entity security limits individual user access to the domains and entities identified in User Domain/Entity Access Maintain (36.3.4). If using the .NET UI, users can open other workspaces in order to access domains and entities for which they are authorized.

These two types of security are closely related and work together to ensure that users can only access the business areas that they have been authorized.

Domain and Workspace Security

Access to domains and entities is controlled at two points:

- During system login
- During the application session

When users start the system (assuming single sign-on is not enabled), they submit user credentials using the login dialog box. See “Single Sign-On Enabled” on page 29 for details on how this setting affects login.

The client authenticates the user by calling the authentication service. If a user’s identity cannot be verified, login to the system fails. This authentication takes place for both the character and the .NET UI login. The system next checks to see if the user has access to any domains and entities defined in User Domain/Entity Access.

This step varies based on the user interface:

- In character, the system checks to see if the user has an assigned domain. If not, an error is generated, and login is refused. If only one assigned domain is found, login to that domain is automatic. A user with access to more than one domain can choose from a list. The one marked as default in User Domain/Entity Access displays at the top of the list.
- In .NET UI, the authentication service creates a session for the user, and returns a session ID to the .NET UI. The .NET UI uses the session ID to initialize any workspaces (domain/entity combination). By default, this is the workspace that was active when the user logged out of a previous session. If no previous session exists, the default domain is used as with character login.

  Note Login to the .NET UI can be successful even when the user is not assigned to a workspace. This is because the .NET UI is a container for multiple applications and also provides access to system administration functions that are not part of any specific application.

Changing domains also differs depending on the UI:

- In the .NET UI, a user with access to more than one domain or more than one entity in a domain switches from one to another by opening a different workspace.
- In the character UI, users switch domains by using Change Current Domain (36.1.1.1.10). This automatically switches to the primary entity of the new domain.

At no time can a user access an entity that is not authorized in their user record. In the .NET UI, these workspaces do not display for selection; in the character UI, attempting to switch to an unauthorized domain or entity displays an error message. See “Specifying Access to Domains and Entities” on page 45.

When a user exits the .NET UI, the active workspace is saved and displays when that user logs in again. In the character UI, the default domain assigned to the user in User Domain/Entity Access always displays by default.

For details about using and managing workspaces, refer to *User Guide: Introduction to QAD Enterprise Applications*.

Login and Security Control

Use Security Control (36.3.24) to define additional security measures related to system login. The measures discussed in this section assume that single sign-on is not enabled.

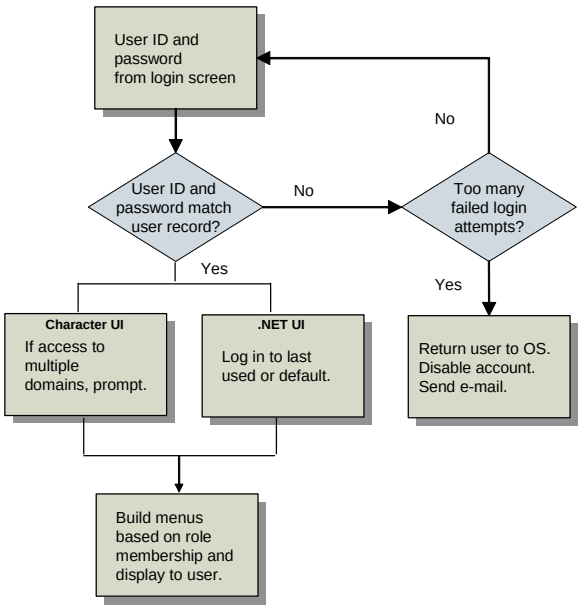
If a user enters an invalid combination of user ID and password, the system may prompt additional times—based on the value of Maximum Access Failures in Security Control. After the specified number of failures, the user is returned to the operating system, the user account is disabled, and system administrators are notified by e-mail. The sending address of the e-mail includes the operating system ID of the user who attempted to access your QAD application. Figure 2.1 illustrates how this process occurs during login. Use the User Account Status Report (36.3.23.2) to view the status of system users.

To completely or partially bypass system login security, you can configure the system to allow users to access the system based on operating system user ID. See “OS-Based Login Security” on page 18.

Depending on the setting specified in Security Control—and if single sign-on is not enabled—the system maintains historical records of successful and failed login attempts. Use Logon Attempt Report (36.3.23.1) to view login history.

Note In order for the time zone to be properly recorded during login and password change, the server time zone must be specified in Database Control (36.24.1).

Fig. 2.1
Login Validation from Login Screen



Using login security, you can:

- Effectively separate QAD application security from the operating system security (unless you choose to control access from the operating system level). The user ID in your QAD application does not have to be the same as the user ID referenced by UNIX or Windows. See “OS-Based Login Security” on page 18.
- Provide an extra level of security from unauthorized users. An individual can gain access to an operating system user ID by breaking into the system or stealing a password. Requiring a different user ID and password combination to access QAD applications presents an additional barrier to an unauthorized user. If single sign-on is enabled, this extra level of security is removed. Consequently, since single sign-on represents a single point of failure, you should consider carefully whether enabling this feature is appropriate in your environment.
- Track unsuccessful login attempts to identify possible unauthorized efforts to access the system.

Single Sign-On Security

For users of the .NET UI in a Windows environment, you have the option of enabling application single sign-on. Single sign-on lets users log in to the Windows environment and start the .NET UI without being prompted for their user credentials, after an initial authentication.

Application single sign-on process works like this:

1. The system administrator enables single sign-on using Security Control (36.3.24).
2. The .NET UI client queries the authentication service (on each login) to determine whether single sign-on is enabled.
   - If single sign-on is not enabled, the .NET UI prompts for the user ID and password.
   - If single sign-on is enabled, the .NET UI determines whether the user's credentials are stored in cache. If so, the credentials are decrypted and authenticated, allowing the user access to the system. If not, the system prompts for user ID and password, then caches the encrypted credentials for use in subsequent login.

When a user changes password, on the next login the system prompts for user name and password; if single sign-on is enabled, it caches the new credentials.

Before choosing to implement single sign-on, you should carefully weigh the advantages of improving the ease of user access against the security considerations of having a single point of failure.

Note If your system users employ the character interface, using single sign-on is not an option—users are required to sign on to their Windows environment and system separately.

For details, see "Defining General Security Settings" on page 26.

OS-Based Login Security

System administrators can control user access to the character interface directly from the operating-system level using the Enforce OS User ID field in Security Control (36.3.24).

If you are not using an application password, using the Enforce OS User ID feature lets you essentially bypass application login security completely and rely on operating-system security for your character-based users.

The .NET UI supports Microsoft's Active Directory authentication for use with the Enforce OS User ID field. With Active Directory support, user passwords can be centrally managed. User accounts must be created in the QAD system, and the User ID must match the Active Directory User ID. Note that in the QAD system, the User ID is limited to eight characters.

Important Regardless of this setting, users logging in through .NET UI must enter a valid user ID and password to access the system.

When the Enforce OS User ID check box is selected, the default user ID displayed in the login screen is the same ID used by the operating system, and the user cannot change it. This must still be a valid system user ID defined in User Maintenance (36.3.1).

In addition, when the Enforce OS User ID check box is selected, the Single Sign-On Enabled option cannot be selected, and vice versa. Enforce OS User ID uses Windows environment variables to verify user credentials. An unauthorized user may potentially be able to reset the %USERNAME% environment variable in order to gain access to the system, masquerading as a different user. You should consider this issue carefully when defining your security model—implementing single sign-on may be a better solution for your environment.

Subsequent processing depends on whether a password is required for the user:

- If no password is specified in the system user record, login proceeds automatically, subject to proper licensing.
- If the user record includes a password, the system displays a password prompt.

Important If you enable this feature and reset user passwords for the application to blank, be careful if the Enforce OS User ID check box is ever cleared. If you do so without reentering passwords in user records, anyone can gain access to the system by entering just a user ID. When you clear this check box, the system displays a message to warn you of a potential security compromise. In addition, if using the .NET UI, it is not recommended that you reset user passwords for the application to blank. It is relatively easy to create a new user on an existing Windows machine with an ID that matches one in the application.

Operating System and Progress Security

Security controls applied using programs on the Security Menu (36.3) apply primarily to accessing the application itself, as well as accessing functions within the application. In addition to system controls, you should consider additional security at the operating system and Progress levels.

At the operating system level, all application-related files should be reviewed to determine the appropriate permission and ownership settings. Relevant files would include at a minimum:

- Database files (*.db)
- Log files (*.lg)
- Source code files (*.p)
- Compiled source code (*.r)
- Database backup files
- Configuration files (*.config)
- Files used to execute system implementation functions such as the QAD deployment tool
- Files that are part of the QAD .NET User Interface

For example, on UNIX platforms, a system administrator should be the owner for most—if not all—of these files. To restrict access to these files, operating system commands such as the following for UNIX can be used to limit both Read and Write access to the file owner.

```
chmod 600 <database file name>
```

The standard Progress documentation set provides information about security controls, including the following documents:

- *Database Administration Guide*
- *Client Deployment Guide*
- *Progress Programming Handbook*

The following sections discuss information-security exposures and mitigating controls in these areas:

- Accessing the Progress Editor from the application
- Capabilities to directly read, modify, and delete database records
- Compiling custom code on unprotected databases
- Accessing an application database directly from Progress

Progress Editor Access

One area of potential security exposure is related to the Progress Editor. Access to the Progress Editor from your QAD application is often essential in troubleshooting technical problems. At the same time, once a user has accessed the Progress Editor, system data can be significantly exposed.

Access to the Progress Editor is available from menu 36.25.80, `mgeditor.p`. You can use roles to limit access to the Progress Editor in the same way as any other application menu programs. Using Role Permissions Maintain, assign appropriate access permissions to the roles you want to be able to access the Progress Editor, and then assign these roles to legitimate Progress Editor users.

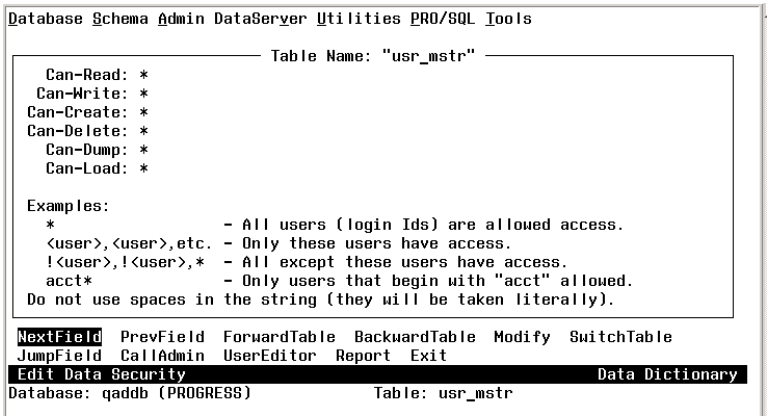
Another related control that should be considered is to disallow privileges for users connecting to the application database with a blank user ID. The Disallow Blank User ID Access option on the Progress Database|Admin|Security menu is available for this purpose. See the “Maintaining Application Security” section in the *Progress Client Deployment Guide* for details.

Selecting this option denies all access privileges to the Progress blank User ID by placing a leading exclamation point (!) in each table and field permission specification for the database. See the next section for details.

Progress-Level Database Schema Controls

Progress-level security controls should also be considered for protecting the application database tables. Progress provides a schema security function to restrict various levels of access to specific database tables. This function is accessed from the Progress Data Administration|Admin| Security| Edit Data Security menu option.

Fig. 2.2
Assigning Schema Controls



Select the NextField option to define access specifications at the individual field level as well.

These access specifications are enforced at compile time. Users are prevented from writing and executing custom source code in the Progress Editor if the code violates access restrictions.

Compiling Custom Code on Unprotected Databases

Progress schema-based controls do not prevent users from compiling code on an unprotected database with no schema-level access restrictions and then executing it on a production database. The schema access restrictions are checked at compile time rather than runtime.

To provide protection against this exposure, consider using the Progress PROUTIL function DBAUTHKEY to set a key for a Progress database. See the *Progress Database Administration Guide* for details.

Once set, this key is embedded in all r-code compiled against the database. In addition, any r-code is checked to verify that it contains this key value before it is permitted to execute. An additional function, RCODEKEY, is available to set or change the key value in specific r-code entries without recompiling source code.

Progress-Level Database Access

Unless properly controlled, it is possible under certain conditions to start a Progress session and then connect to an application database without starting the application itself. After connecting, there would be no effective controls over accessing private or confidential data, modifying, or deleting records. Since an application session is never initiated, any application-level controls such as menu security could be circumvented. To mitigate this exposure, user and password access controls can be implemented at the Progress level as well as the system level.

To set Progress security, access the Edit User List option on the Admin|Security menu of the Progress Data Dictionary. Use this function to load valid user ID, name, and password combinations into the user security (_user) table.

Note Controls on user IDs and passwords that have been implemented for the application do not apply to user records in the Progress _user table.

You can use this table in combination with command-line security options when the database is started. There are several possibilities:

1. No Progress users are defined and the –U and –P options are not specified. This is the default. The Progress user ID is set to the operating system login or the network logon ID.
2. Progress users are defined but the –U and –P options are not specified. On all systems, this results in a blank Progress user ID. This can be used to establish basic system security for the majority of users. Any users with additional capabilities must specify a –U and –P at startup.
3. Progress users are defined and the –U and –P options are specified. The system verifies that the user ID and password combination is in the user security (_user) table. If not, an error displays and the session is not started.

Note If no Progress users are defined, the –U and –P options cannot be specified.

By setting Progress user/password controls on the application database, restricting access to the database files, and monitoring the database log file for unusual access events, security exposures from inappropriate access to the application database can be substantially reduced.

Workstation-Level Security

Depending on the operating system of the machines that are running application sessions, you may be able to combine an application security setting with operating system features to create an additional security layer at the workstation level.

The Timeout Minutes field in Security Control (36.3.24) lets you specify the number of minutes of inactivity that can occur before the system automatically logs a user out of an application session. Primarily used to reduce the system load resulting from users who stay logged in when they really do not need to be, this feature also enhances access security. If you set this to a reasonable number—such as 30—you can prevent users from inadvertently staying logged in when they go to lunch and leaving an open session that might be accessed by unauthorized individuals. See the section “Timeout Minutes” on page 26.

In the character UI, this feature applies only when the application is displaying a menu, rather than when a program is executing. In the .NET UI, time out is applied regardless of whether the user is displaying a program screen.

Note To add workstation security for times when a user leaves a computer unattended while a program is running, you can use operating system features.

Windows Systems

In many environments, users run the application on a Windows system; for example, character sessions using a terminal emulator, or the .NET UI. You can establish work procedures that require users to set up their machines to display a screen saver after a specified number of minutes and enter their Windows password—preferably not the same one used for the application login—to turn off the screen saver.

Note This procedure assumes that users require passwords to access their computers.

1. Right-click the Windows desktop.
2. Select Properties.
3. Click the Screen Saver tab.
4. In the Wait field, enter the number of minutes that the machine is idle before the screen saver displays.
5. Select the box labeled On resume, password protect.
6. Click OK.

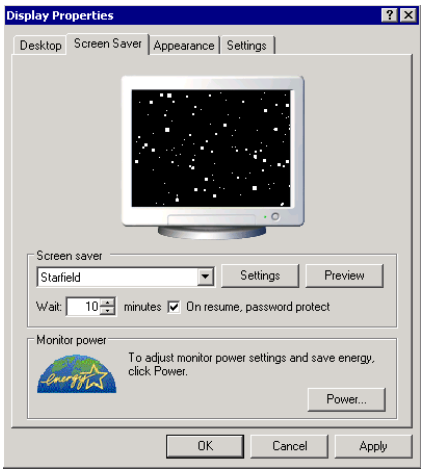
When the screen saver comes on, it can be cleared only when the current user’s Windows password is entered, or when an individual with system administrator access overrides the user login.

Note Setting up this form of security does not affect any applications that are running when the screen saver displays—it only blocks access to the computer.

For details, refer to your Windows system documentation.

Figure 2.3 illustrates an example of a computer running Windows XP set up for a 10-minute screen timeout, which can be cleared only by entering a password.

Fig. 2.3
Example of Windows Screen Saver Setup



To lock a computer manually without waiting for the screen saver timeout, press Ctrl+Alt+Delete, then click Lock Computer. A password is required to access a locked system. Your security policy should require users to do this when they leave their computers unattended as a matter of good security practice.

Note Depending on the operating system and version running on your Windows computers, as well as the way users are set up, the system administrator may be able to configure all machines in this manner and prevent individual users from changing the settings. Refer to your operating system documentation for details.

Non-Windows Systems

Many standard UNIX workstations—including those provided by HP, Sun, and IBM, which use the Common Desktop Environment (CDE)—offer screen-locking features much like those in Windows. Set up CDE-based machines using the Style Manager icon on the Front Panel. Similar features are also available for some LINUX environments. See the user documentation for your workstation for details.

.NET UI Security

The .NET UI supports external customized menus defined in XML. The ability to customize menus allows you to add content—the QXtend plug-in, for example—outside of standard programs and functions.

The items on external menus are not filtered unless a security constraint is added to the menu; this is achieved by manually editing the menu extension configuration file. Security constraints can be placed on the XML file by user or role.

For details, refer to the installation guide.

Chapter 3

Setting Up Security Control

This section discusses how to set up basic security in your system.

Defining General Security Settings

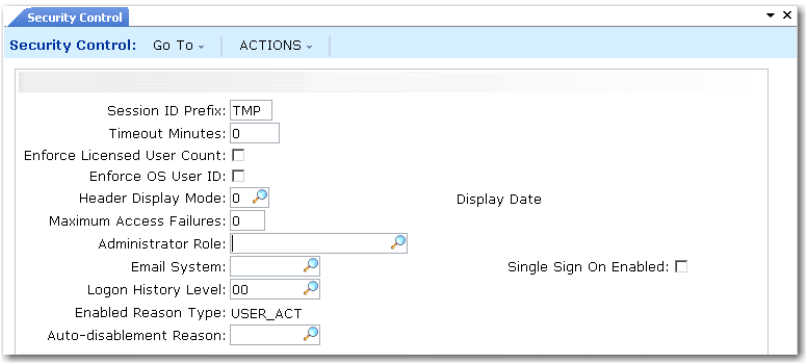
Use the two frames of Security Control (36.3.24) to:

- Establish basic security parameters for your environment
- Define the way you want to set up and control passwords

Two special security considerations apply to records created in this program:

- Whenever a field is updated, the system notifies administrators by e-mail. See “Setting Up E-mail Notifications” on page 31 for details.
- You must use this program to update data values in the user control (usrc_ctrl) table. The system prevents you from using other methods, such as the Progress Editor, to modify that record.

Fig. 3.1
Security Control (36.3.24), Initial Frame



Session ID Prefix. This field is no longer being used. Session IDs generated in the .NET UI use a complex algorithm that ensures uniqueness without a prefix.

Timeout Minutes. Specify a number of minutes after which the system should automatically log out inactive sessions. Set a value in this field to minimize unnecessary overhead on busy systems.

Note If a nonzero value is entered in this field, the Timeout daemon must also be configured and started. For details, refer to *User Guide: QAD System Administration*.

The field also can be used as part of an overall security strategy to prevent users from inadvertently allowing access to unauthorized individuals. See “Workstation-Level Security” on page 21 for details.

If you enter a value, when the system considers a session inactive depends on the UI:

- In the character UI, the time out is applied only when a menu is displaying, such as Item Data Menu (1.4) If the user is executing a program—Item Master Maintenance (1.4.1), for example—a session is never automatically logged out.
- In the .NET UI, the time out is applied regardless of what the logged-in user is doing. This is because the load on system resources for inactive users is much greater in the .NET UI.

Enforce Licensed User Count. Use this option to implement enforcement of the total number of users, sessions, or transactions allowed based on your license agreement.

Not selected (the default): The system issues license violation warnings if you violate your license agreement, but you are not prevented from completing the action that caused the violation.

Selected: The system issues a violation error if you violate your license agreement and you cannot complete your current activity.

The system tracks all license violations, both warnings and errors. License violations can occur in the following situations:

- In User Maintenance (36.3.1) when you attempt to add users or assign them to applications
- In License Registration (36.16.10.1) when you assign users to applications
- During user login to the system
- When users attempt to use separately licensed applications or nonregistered applications

Important Violation warnings should not occur often; if repeated warnings occur, contact your QAD representative or distributor for a license upgrade.

Enforce OS User ID. Specify whether the system allows users to access character sessions for the application based on their operating system login. See “OS-Based Login Security” on page 18 for details.

Not selected: Users must always enter a valid user ID and password.

Selected: Depending on password parameters defined in Security Control, valid users defined in the system may be able to access the application directly without entering login information.

Note The Enforce OS User ID option cannot be selected if the Single Sign-On Enabled option is selected.

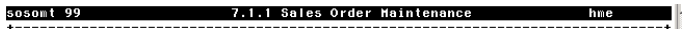
Header Display Mode. Use this field to control the information that displays in the menu and program title bars of programs in the character interface. Valid values are:

0 (Display Date). The menu title bar displays the name associated with the current domain followed by the current database name defined in Database Connection Maintenance (36.6.1). The program title bar from left to right includes the program name, the version of the program, the menu number and title, and the current date (see Figure 3.2).

Fig. 3.2
Header Display Mode 0

1 (Display User ID). The menu title bar is the same as choice 0. The program title bar is the same as choice 0 except that the login ID of the current user replaces the current date. Reading from left to right, the title bar includes the program name, the version of the program, the menu number and title, and the login ID of the current user (see Figure 3.3).

Fig. 3.3
Header Display Mode 1



2 (Display Date and Domain). The menu title bar displays only the current database name defined in Database Connection Maintenance. The program title bar from left to right includes the short name and currency of the current working domain, the menu number and title, and the current date (see Figure 3.4).

Fig. 3.4
Header Display Mode 2

3 (Display User ID with Domain). The menu title bar is the same as choice 2. The program title bar is the same as choice 2 except that the login ID of the current user replaces the current date. Reading from left to right, the program title bar includes the short name and currency of the current working domain, the menu number and title, and the login ID of the current user (see Figure 3.5).

Fig. 3.5
Header Display Mode 3

Some regulatory environments may require the name associated with the user ID of the logged-in user to be available from any program. In the character interface, you can use the Ctrl+F key combination to review this information and other context details.

Maximum Access Failures. Enter the maximum consecutive failed login attempts allowed before the system disables the user's login ID. When an account is disabled, the system sends an e-mail message to the system administrator. See "Setting Up E-mail Notifications" on page 31 for details.

Leave this field set to zero (0) if you do not want to limit failed access attempts.

Note If you are using electronic signatures, this same value controls the number of failed signature attempts that are allowed before the system disables the user ID.

Administrator Role. Specify the role assigned to system administrators. The members of this role receive e-mail notifications when specific security and controlled events occur; for example:

- When a user account is disabled for too many failed login attempts. See page 31 for details.
- If you are using electronic signatures, when an electronic signature profile is activated or a user account is disabled for too many failed signature attempts.
- When an update is made in Security Control. See page 31 for details.

Typically, the administrator role includes a primary system administrator and one or more alternates.

Email System. Specify an e-mail system definition—set up in E-Mail Definition Maintenance (36.4.20)—used to notify system administrators when security- and internal control-related events occur.

Note The system first attempts to use the e-mail definition specified for the logged-in user in User Maintenance. If the user record does not include a valid e-mail definition, the one specified in this field is used. See *User Guide: QAD System Administration* for details on setting up e-mail.

Logon History Level. Indicate the level of system-maintained login history.

None (the default): Login history is not maintained.

Failed: Login history is maintained only for failed login attempts.

All: History is maintained for all login activity.

Particularly in highly regulated security environments, you can use login history information as part of an overall access monitoring effort. Use Logon Attempt Report (36.3.23.1) to view login history. See “Monitoring System Security” on page 32.

Note Be sure to set this field based on the level of information you think will be needed when you run the report. For example, if you set the history level to None, Logon Attempt Report will not include any data.

Enabled Reason Type. This is a display-only field. The system-assigned value is USER_ACT, the reason type associated in Reason Codes Maintenance (36.2.17) with reason codes used by security functions. The system uses reason codes of this type in two places:

- The Auto-Disablement Reason field.
- Reason codes entered manually in the Enabled Reason field in User Maintenance. See “Enabled Reason” on page 43 for details.

Example You could use Reason Codes Maintenance to create the following reason codes associated with type USER_ACT:

- AUTO. The system automatically disabled the account. You could enter this in Auto-Disablement Reason.
- REACT. The system administrator has manually re-enabled the account.
- NEW. The system administrator has added the account for a new user.
- LEFT. The user is no longer with the organization, and the system administrator has disabled the account.

Note System installation or conversion automatically creates one default reason code, QAD_DEF, for reason type USER_ACT. After installation, this code displays in the Enabled Reason field in the user record of the default system user, mfg. During conversion, existing user records are populated with this value. After you set up values in Reason Codes Maintenance that apply to your system, you do not have to use this default reason code.

Auto-Disablement Reason. Enter the reason code the system enters in user records when it automatically disables a user account. This occurs when the user reaches the number of consecutive failed login attempts specified in Maximum Access Failures. This code must be defined in Reason Codes Maintenance and be associated with reason type USER_ACT.

Important Reason codes are domain specific. During security planning, you should determine the codes you will use and set them up as part of the system domain. This way they are copied by default to all new domains.

Single Sign-On Enabled. Specify whether the system allows users to enter their user ID and password once to log in to the operating system without entering additional user credentials to log in to a .NET UI application session.

Not selected: Users are always required to log in to their Windows environment and the application separately by entering user credentials for both.

Selected: Users can log in to the operating system and the application by entering user credentials once when logging in to the operating system.

Note The Single Sign-On Enabled option and the Enforce OS User ID option cannot be selected at the same time.

For details see “Single Sign-On Security” on page 17.

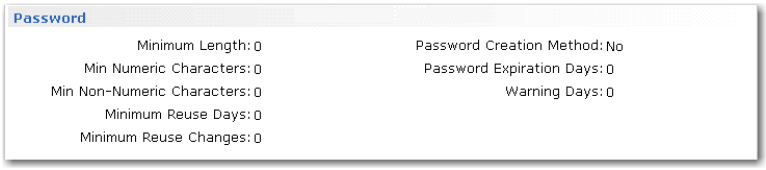
Creating a Password Strategy

Use the Password frame to define the complexity requirements and expiration time period for user account passwords. Anytime a new password is created for an account—either manually or automatically—that password must meet the rules you set up here. Use as many or as few password parameters as required by the security guidelines set for your environment.

If you enable automatic password creation by setting Password Creation Method to Email or Display, the system uses the parameters you specify to generate new passwords.

If you choose to allow valid users to access the application based directly on operating system security, do not define any password parameters; select the Enforce OS User ID check box in the initial frame of Security Control. To default the user ID from the operating system but still require a password for the application at login, select the check box and specify password parameters as needed. See “OS-Based Login Security” on page 18.

Fig. 3.6
Security Control, Password Frame



Minimum Length. Enter the minimum number of characters allowed for new passwords. Password cannot exceed 16 characters. Leave the default 0 (zero) to indicate that a blank password is allowed.

Note Passwords are validated against structure requirements only when they are first created, rather than each time they are used. To make password structure changes apply immediately, use Force Password Change Utility (36.3.23.12) to force users to change their passwords at the next login. New passwords must meet the updated structure requirements. See “Monitoring System Security” on page 32.

Min Numeric Characters. Enter the minimum number of numeric characters required for new passwords. This value plus the value in Min Non-Numeric Characters cannot exceed 16 and must be the same as or less than the specified minimum length. Leave the default 0 (zero) to indicate that numeric characters are not required in the password.

Min Non-Numeric Characters. Enter the number of non-numeric characters required for new passwords. This value plus the value in Min Numeric Characters cannot exceed 16 and must be the same as or greater than the specified minimum length. Leave the default 0 (zero) to indicate that non-numeric characters are not required in the password.

Minimum Reuse Days. Indicate the number of days a user must wait before a password can be reused. The system maintains all user passwords for historical purposes. If users define new passwords at specific time intervals, you can set this value so that the same password is not reused for a specific period of time.

Example Enter 364 to indicate that users cannot select a password already used in the previous year.

This password check can be used independently or in conjunction with the next field, Minimum Reuse Changes. If you set both options, both rules apply. Leave the default 0 (zero) to indicate that this rule should not apply.

Minimum Reuse Changes. Indicate the number of password changes required before a password can be reused. The system maintains all user passwords for historical purposes. You can set this value so that the same password is not reused until the user has changed their password at least this many times.

Example Enter 3 to indicate that users must change their passwords three times before they can use the same password again.

This password check can be used independently or in conjunction with Minimum Reuse Days. If you set both options, both rules apply. Leave the default 0 (zero) to indicate that this rule should not apply.

Password Creation Method. Specify the method you want to implement for creating new temporary passwords. For details on password maintenance, see “Updating Passwords” on page 43.

- No (the default). The system administrator must define temporary passwords manually. Automatic password generation is not enabled.
- Display. A new temporary password is automatically generated and displayed on the screen in User Maintenance. The system administrator must then communicate it to the user.
- Email. A temporary password is automatically generated and e-mailed to the address defined in User Maintenance for the user ID. This method is especially useful in high-security environments because the user is the only person who has access to the temporary password. See “Setting Up E-mail Notifications” on page 31.

Note All passwords created using the specified method are temporary, single-use passwords. The user is forced to change this password at the first login.

Expiration Days. Specify the number of days users can use the same password before the system prompts them for a new one.

Once the specified number of days passes since a user’s last password change, they are prompted for a new password at the application welcome screen. When this field is 0 (zero), passwords never expire.

Note The date of the user’s last password change displays in User Maintenance and User Password Maintenance.

Warning Days. Enter the number of days before a password will expire when users are warned of the upcoming expiration date. This must be less than the value of Expiration Days.

Users are reminded of the expiration date at each subsequent login and can optionally update their passwords immediately or, depending on menu access, update them in User Password Maintenance.

Setting Up E-mail Notifications

Based on Security Control settings, the system can automatically send e-mail to users in the following security-related situations:

- When a user’s consecutive number of failed login attempts exceeds the number specified in Security Control, the system generates and sends e-mails to members of an administrator role. The e-mail text is similar to the following:

```
The purpose of this email is to inform you that a user has been disabled for exceeding
```

```
the maximum logon failures allowed as setup in Security Control. You have been included
in this email distribution because you belong to the Administrator role identified in
Security Control.

User ID disabled for exceeding max logon failures allowed: User ID

This e-mail was automatically generated from an application process. If you have any
questions about this e-mail, contact the system administrator. Do not reply to this e-
mail.
```

- When Password Creation Method is set to E-mail in the Password frame of Security Control, the system generates a new password and e-mails it to the user based on the e-mail address specified in User Maintenance. This occurs for new and existing users when the Update Password check box is selected in User Maintenance. The e-mail text is similar to the following:

```
The purpose of this e-mail is to inform you of your new temporary application password.
You have been sent this e-mail because Security Control has been set up to e-mail
autogenerated temporary passwords.

Your temporary application password is: password.
You will be forced to change this password at next logon.

This e-mail was automatically generated from a QAD process. If you have any questions
about this e-mail, contact the system administrator. Do not reply to this e-mail.
```

- When any field or check box is updated in Security Control, the system generates and sends e-mails to members of the administrator role. The e-mail text is similar to the following:

```
The Security Control menu program has been used to change the security configuration of
QAD. Please review this information carefully to ensure that these changes will not
compromise the system security. You have received this email because you are an
Administrator identified in Security Control for QAD.

Changes made by user: jnw

Changed Field: old, new
======================
Administrator Role: 200401170000219243.4321, 200312090000112641.4321
Password Expiration Days: 99, 0
Logon History Level: 2, 1
Maximum Access Failures: 99, 0
Header Display Mode: 1, 2
Enforce OS User Id: yes,

This email was automatically generated from a QAD process. If you have any questions
about this email, contact the QAD system administrator. Do not reply to this email.
```

  Note Values shown in this message are those stored in the database and may not be the same as displayed in the user interface. For example, the Administrator Role values display as the unique object identifier (OID) codes associated with the old and new values in the database. The message is intended primarily to show administrators which fields were changed.

Monitoring System Security

Particularly in environments where security procedures are subject to regulatory controls, system administrators need methods of tracking security-related events.

The system provides automatic features to help administrators control and monitor security activities:

- Based on settings in Security Control, users who enter an incorrect user ID/password combination more than a specified number of times are automatically locked out of the system. They can use their user ID again only after the system administrator has reenabled it.
- When an account is disabled, the e-mail system can automatically notify system users that have been assigned an administrator role. This serves two purposes:

  - In cases where the user simply forgot a password or mistyped it repeatedly, the administrator can quickly restore access.
  - The administrator knows immediately if an unauthorized user is attempting to access the system with a known user ID. This lets the administrator take appropriate steps such as immediately requiring all users to change their passwords. Force Password Change Utility (36.3.23.12) lets the administrator force users to update their passwords based on role, domain, and/or the date of the last change.
- Depending on the level of login history specified in Security Control, use Logon Attempt Report (36.3.23.1) to track when login attempts take place. This could be useful, for example, to track specific times when unauthorized users are attempting to access the system. The report shows such information as the user ID of the person who attempted the login, as well as the date, time, server time zone, and other data relevant to the login event. (If you are using electronic signatures, E-Signature Failure Report (36.12.7) lets you monitor unsuccessful signature events.)

  Example You can set up batch processing to run this program each morning to identify all failed login attempts on the previous day.
- Each time a user account is enabled or disabled, the Enabled Reason field in User Maintenance must be updated. This happens automatically when an account is disabled as a result of excess unsuccessful login attempts. Otherwise, the administrator must enter a reason code manually.

Chapter 4

Setting Up Users and Roles

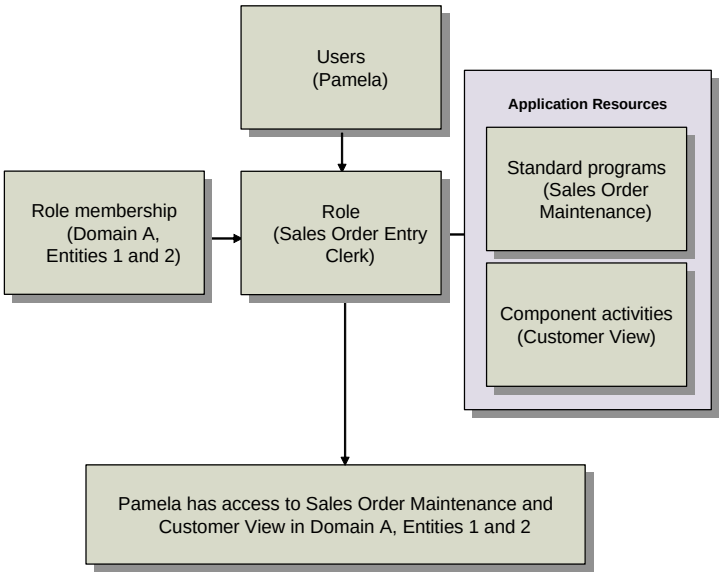
This section describes how to set up users and roles in your system.

Overview

Role-based access control is a security mechanism that is designed to work with two basic user-defined elements: users and roles. Role-based access control limits users to executing only the system menu items belonging to their assigned role or roles.

Figure 4.1 illustrates the interaction of system users, role permissions, and role membership to determine the resources that are available to a user.

Fig. 4.1
Users and Roles

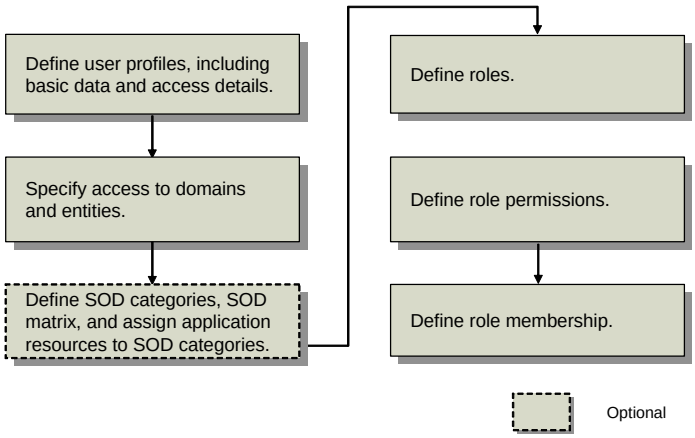


Role and User Definition Process Workflow

Before implementing your security model, you should develop a detailed security plan that describes how users and roles will be defined within your system to satisfy the business requirements of your organization. For details, see “Implementation Summary” on page 5.

Use the programs in the System Security Menu to set up and configure users and roles in your system. Figure 4.2 shows the user and role setup process workflow.

Fig. 4.2
Users and Roles Setup Flow



1. Create system users in User Maintenance (36.3.1). This step identifies each user to the system by providing them with a unique ID. You also provide basic user information to ensure that system data for each user is correctly displayed and processed, as well as specify security-related access settings and licensed applications. For details, see “Setting Up Users” on page 37.

2. Specify user access to domains and entities in User Domain/Entity Access Maintain (36.3.4). For details, see “Specifying Access to Domains and Entities” on page 45.

3. If you plan to implement segregation of duties, it is best to implement this internal control prior to defining roles and role permissions. Once associations between application resources and SOD categories have been defined, role permission definitions are constrained by your SOD policy. Implementing segregation of duties is optional.

   Note Segregation of duties is not supported in this initial release.

4. The next required activity is to create roles in Role Create (36.3.6.1). All system users must be assigned to a role before they can access the system. For details, see “Setting Up Roles” on page 47.

5. After creating user roles, define role permissions using Role Permissions Maintain (36.3.6.5). Role permissions determine which menu-level programs and activities a user can execute; they also determine a small number of non-menu level permissions. For details, see “Defining Role Permissions” on page 52.

6. Then use Role Membership Maintain (36.3.6.6) to assign users to roles and specify the role context—that is, how the role operates within domains and entities. For details, see “Defining Role Membership” on page 55.

Setting Up Users

The process of setting up users identifies the users to the system and defines user-related information that the system requires. This process consists of:

- Defining users including:
  - Basic user information

  - Security settings
  - Application use
- Specifying the domains and entities each user can access

Types of Users

One of the fields that you specify when you create a user indicates the user type. Most users represent your company employees that perform day-to-day functions such as receiving purchased inventory, replenishing work centers, and filling sales orders.

However, the system also requires a number of users for performing background tasks that require system login. These users do not represent real individuals, and are typically given a user type of API (application program interface). Generally, this type of users should be associated with a role that grants full access to all domains, entities, and resource so that the required background tasks can be performed.

You specify these types of users in a number of different places:

- All of the daemon processes require a valid user ID and password for logging into the system. Typically you should create one user with access to all domains, entities, and resources and specify the same user for all the daemons. This makes administration simpler.
- System Maintain (36.24.3.2) requires a user ID and password for system startup activities that are initiated from the operating system or from a shortcut. This ID is used to establish that a valid user session can be created.
- A user role is defined during installation for .NET UI administrative functions, that again needs access to all system resources.
- If you are using other components of QAD Enterprise Applications such as QAD Customer Self Service or QAD QXtend Inbound or Outbound, you need to configure a special user for interaction between the components.

Defining Users

Use User Maintenance (36.3.1) to assign a unique ID to a system user and define related application and security details.

To access the system, each user must specify a unique user ID and the associated password. In addition each user must have been assigned a valid role and access to one or more domains and entities. Other user data is referenced throughout the system and may be required for reasons other than security.

User profiles apply to all domains in the system. However, profiles include several generalized codes that are domain specific such as access location and user type. To prevent validation errors, you should ensure that these codes exist in all domains.

Once a user has accessed the system, the ID cannot be deleted. Instead, you can deactivate a user’s record in the system. If an ID has never been used for login, you can delete it, if necessary. This lets you correct any errors made during initial setup. This restriction ensures a complete audit trail of users who have accessed the system.

Important The Active check box and the Enabled check box in User Maintenance have different functions.

- The Active check box controls whether a user's record is active within the system. Only active user records can be referenced when a new record is created in other system functions; in addition, lookups and browses only display active records.
- In contrast, the Enabled check box determines whether a user can log in. By default, the Enabled check box is selected when a new user is created. A user can log in to the system only if both the Enabled check box and Active check box are selected. The account of an active user can be disabled, for example, while they are on medical leave.

Fig. 4.3
User Maintenance (36.3.1)

User ID. Enter a code (maximum 8 characters) identifying a user in this database. This field cannot be blank or the same value as a role name. Do not use an exclamation point (!) or comma (,).

To log in to the system, the user must supply a valid user ID.

If you plan to use OS-based security, the user IDs you create should be the same as the IDs defined for operating system login. See "OS-Based Login Security" on page 18.

Depending on the setting of Header Display Mode in Security Control (36.24), the system may display this value on every program screen in the character interface. In the .NET UI, the user ID always displays in the bottom message area. See "Header Display Mode" on page 27.

User Name. Enter a user name (maximum 35 characters) identifying the full user name associated with this ID.

The user name does not affect system security. It displays for reference on various reports and inquiries. To display an information window that includes the user name, press Ctrl+F from any program screen in the character interface.

Defining Basic User Information

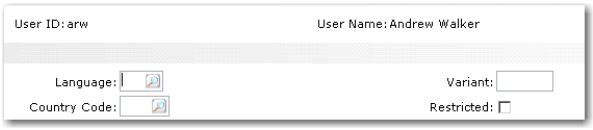
Defining basic information about system users includes setting options and defining values for:

- Controlling information process and display
- Identifying users
- Specifying e-mail addresses
- Enabling menu substitutions

Controlling Information Process and Display

You can ensure that system data is correctly displayed and processed for a given user—regardless of the user's language or location—by specifying values for the Language and Country Code fields in User Maintenance.

Fig. 4.4
User Maintenance, Language and Country Code



Language. Enter a two-letter code identifying the user's language. The system displays menus, messages, and other interface elements in this language when the user logs in.

The language must be active and must be installed. Since labels, menus, messages, comments, and field help text are stored and retrieved by language code, you cannot assign a language to a user when these elements have not been loaded. Loading translated data automatically sets the associated language to installed.

Changes to this field do not affect any users currently logged in. Changes take effect only when they log in again.

Country Code. Enter a valid, active country code defined in Country Create (36.1.3.1.1). The country code also must have an associated alternate country code defined in Country Code Data Maintenance (2.14.1).

The alternate country code must be a valid International Standards Organization (ISO) country code. The system uses the ISO code to set up date and number formats and other interface elements for each user session.

Variant. Optionally enter the locale for the user. This field can be used to specify regional variations within a country.

Information on language, country code, and variant are maintained in a file named `locale.dat`, along with other format information. Once the system determines a user's language, country code, and corresponding ISO country code, it gets information from `locale.dat` and uses it to set user-specific date and number formats. See the installation guide for more information.

System administrators may need to change information in `locale.dat` or add entries for countries that are not included in the current file.

Each line in the file follows the same format. For example, the line for US English looks like this:

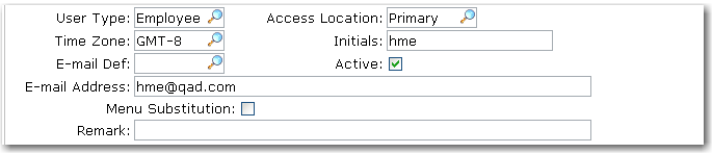
```
US,en,US,,mdy,American
```

Where:

- US is the application language code.
- en is the ISO language code.
- US is the ISO country code.
- Optional variant is blank.
- mdy is the date format.
- American is the numeric format (period as the decimal separator; comma as the thousand separator).

Identifying Users

Fig. 4.5
User Maintenance, User Identity Fields



Use the following fields to identify this user:

User Type. Specify the type associated with this user.

- Employee identifies internal users who are employees.
- Customer identifies external customers who are authorized to access the system remotely. To assign a customer type to a user, you must enter a valid customer ID as the user ID in User Maintenance.
- QAD identifies QAD employees who do customer support or service work.
- API identifies users who access the system through an application programming interface connection or who represent background processes such as daemons.

Employee is the default for all newly created users except customers. When you enter a customer ID as the user ID, the type defaults to customer.

You might need to define additional types if users do not fit into the four categories; for example, you may need a contractor or part-time type. You must predefine the new user type in Language Detail Maintenance (36.4.2) before you can assign it to users here.

Time Zone. Enter a time zone to associate with this user. Time zones must be predefined in Multiple Time Zones Maintenance (36.16.22.1). Time zone defaults from the server time zone specified in Database Control (36.24.1).

Access Location. Enter a code that associates the user with a major business facility or major business location. If you have more than one facility or location or if users work remotely or in small offices, associate the user with the major business facility or location that is most appropriate.

Access location codes must be defined in Generalized Codes Maintenance (36.2.13) for field usr_access_loc. The system ships with a Primary location code that is used as the default for new user records. You can use this location as your company home office location or central processing site.

Initials. Enter initials for the user (maximum 20 characters). Initials can be used in references and when performing searches.

Active. Indicate if this is an active record.

When a record is active, it can be referenced from other maintenance functions. When a record is inactive, it cannot be referenced when a new record is created in other functions. Inactive records are not included in lookups of valid values. However, marking a record as inactive does not prevent you from continuing to use existing records that reference the inactive value. In addition, inactive values display on reports.

Once a user ID has been used for login, it cannot be deleted from the system. If an ID is no longer needed, deactivate it.

The system automatically selects this check box for new users.

Remark. Enter a brief text comment regarding the user. For example, you could note that this user is currently on leave of absence and the ID has been disabled.

Specifying E-Mail Addresses

Associate a valid e-mail address and definition with each user who receives system-generated messages by entering values into the E-Mail Address and E-Mail Definition fields.

E-mail can be used with many system features. For example:

- System administrators can receive automatic notification when user IDs are disabled because of login violations.
- Based on a Security Control setting, users can receive system-generated passwords by e-mail.

  Note If you plan to use this feature, be sure to specify e-mail data when you set up user accounts so that users can receive their passwords.
- Various internal control features, such as segregation of duties and e-signatures, use e-mail to inform administrators of unusual system events.

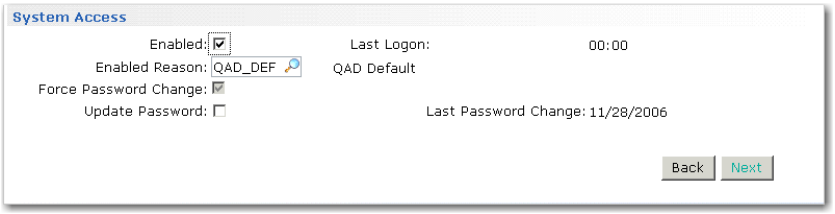
Enabling Menu Substitutions

Select the Menu Substitution check box to indicate whether menu substitution is enabled for individual users when employing the character interface. When menu substitution is enabled, inquiries display instead of browses. This setting has no effect when using the .NET User Interface.

Specifying Security Settings

Use the System Access frame in User Maintenance to specify security-related access settings for each user.

Fig. 4.6
User Maintenance, System Access Frame



Enabled. Select the check box to indicate that this user ID can be used to log in to the system. To disable an existing user ID, clear the check box.

The Enabled check box has a different function than the Active check box. The Enabled check box controls the ability of a user to log in to the system. In contrast, the Active check box controls whether a user's record is active within the system.

Note Any time this check box is updated, the Enabled Reason field must also be updated.

Enabled is updated in the following ways:

- Automatically when you enter a new user ID. By default, the system selects the Enabled check box; you must manually enter an enabled reason.

- Automatically when the system disables an account for too many failed login attempts. Enabled Reason is set to the code specified in Security Control. See “Maximum Access Failures” on page 28.
- Manually when you update an existing ID; for example, you can do this to re-enable a user that was previously disabled, or to disable an account when a user leaves the organization. You must enter an enabled reason.

Enabled Reason. Enter a reason code that indicates the reason for modifying the setting of Enabled. This reason code must be associated with reason type USER_ACT. See “Enabled Reason Type” on page 29.

You must update this field anytime you change the Enabled field.

Force Password Change. Indicate whether the system should force this user to create and validate a new password the next time they log in to the system using the current password.

By default, the system selects this check box for new users and the check box cannot be updated. This lets you assign temporary, single-use passwords either automatically or manually.

By default, the system clears this check box for existing users unless the password has been changed. In that case, it is automatically selected and you cannot update it. This forces users to assign their own passwords at the next login.

Use Force Password Change Utility (36.3.23.12) to select this check box for users belonging to selected roles.

Update Password. Specify whether this user requires a new password. For new users, the system selects this check box by default, and you cannot change it.

Updating Passwords

When the Update Password check box is selected in the System Access frame, subsequent actions depend on the setting of Password Creation Method in Security Control:

- Display. The system-generated password displays at the bottom of the screen.
- Email. The system generates a password and e-mails it to the user.
- No. Automatic password generation is disabled. A frame displays for you to manually enter a new password.

Note Passwords specified in User Maintenance are single-use, temporary passwords generated by the system or entered by the system administrator. At login, the user is prompted to enter a new password.

Fig. 4.7
User Maintenance, Set New Password Frame

Enter a new password. Since the system does not display passwords, type it again to confirm it.

Note The new password must conform to structure and reuse rules defined in Security Control.

Passwords expire based on the value of Expiration Days in Security Control. If you want to let users change their own passwords at a time other than login, give them access to User Password Maintenance (36.3.3). See “Expiration Days” on page 31.

Specifying Application Use

QAD applications support a number of license types. If you are using named user licensing, a finite set of users is predefined.

When the user count exceeds the number of licensed users, a violation message displays here. Violation messages can be either warnings or errors, depending on whether enforcement of the license policy is implemented or not. This is determined by the setting of Enforce Licensed User Count field in Security Control. See “Enforce Licensed User Count” on page 26.

- When Enforce Licensed User Count is Yes, an error displays and you cannot add new users when user count exceeds the number of licensed named users.
- When Enforce Licensed User Count is No, a warning displays and a violation is recorded, but system administrators can add new users.

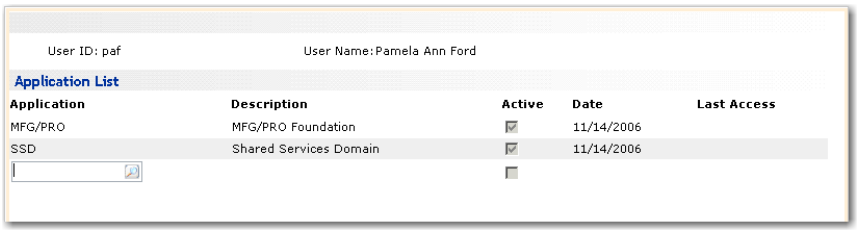
Important After you receive a warning, you can continue with software use. If you receive repeated warnings, contact your QAD sales representative or distributor for a license upgrade.

The applications that a user can access must be activated for the user; otherwise, the user cannot access the application. You can activate access to applications here, or when you register an application license code in License Registration (36.16.10.1).

Once a user has accessed the system, the ID cannot be deleted. Instead, you can make users inactive for an application. If an ID has never been used for login, you can delete it, if necessary. This lets you correct any errors made during initial setup.

Use the Application List frame in User Maintenance to define the software applications that a user can access. When you define a new user, the system prompts you to authorize the new user for all licensed applications. If you select the check box, the Active check box is selected for all licensed applications for this user. Otherwise, Enterprise Applications (MFG/PRO) is listed as the only active application. You can list additional licensed software applications, then select (or clear) the Active check box for each application. By default the check box is selected.

Fig. 4.8
User Maintenance, Application List Frame



The application name you enter under Application Name must be registered with the system through License Registration (36.16.10.1). If not, an error message displays.

The system counts the number of enabled users authorized to access the application and compares the number against a predefined limit for your license type. If the number of enabled users exceeds the predefined limit, a violation message displays and you cannot add the application to the list.

You also can specify which users can access an application after you register the application in License Registration.

If you disable the Enterprise Application (MFG/PRO) setting for a user, all other registered applications are also disabled.

Use User Access by Application Inquiry (36.3.22) to view a list of applications as well as the user's ID and name, active or inactive status of each application, time zone, access location, and access date.

Fig. 4.9
User Access by Application Inquiry (36.3.22)

Specifying Access to Domains and Entities

Use User Domain/Entity Access Maintain (36.3.4) to create or maintain user access privileges for domains and entities. The combination of domain and entity represents a workspace in the .NET User Interface.

If you specify more than one domain, identify the default domain that the system should display at login in the character interface.

To view access privileges for domains and entities, use User Domain/Entity Access View (36.3.5).

Function Overview

You must always define access to a combination of domain and entity. However, the entity dimension applies largely to financial data; most operational functions do not directly update data that is maintained at the entity level. When you are setting up users in a domain with multiple entities and these users will be working exclusively in operational areas such as manufacturing, sales, or service, assign them to the primary entity.

Be aware, however, that certain operational functions such as Operational Transaction Post (25.13.7) and Invoice Post and Print (7.13.4) do update entity-specific data. In these programs, access security defined in User Domain/Entity Access Maintain determines which entities can be updated.

Note The level of security access enforced—either domain or entity—displays in Role Permissions Maintain next to each menu item.

Domains and entities are defined as part of the process of setting up your foundation data. Refer to Chapter 3, "Setting Up Financial Foundations," in *User Guide: QAD Financials*.

New domains and entities that are added to the system after implementation display in User Domain/Entity Access Maintain. You must explicitly grant users access before any updates can be made in the new domain.

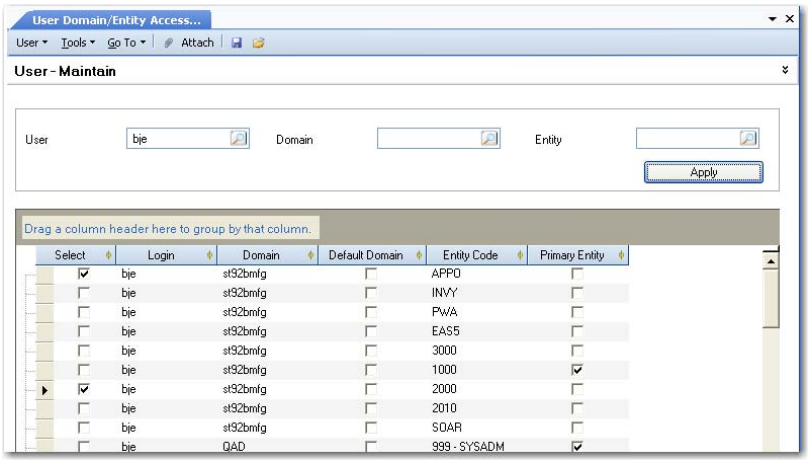
Any changes to a user's domain or entity access privileges automatically update that user's role membership information. For example, removing a user's ability to access an entity breaks the association between that entity and the user's assigned role, and the entity is deleted from the list of assigned entities in Role Membership Maintain (36.3.6.6.1). For details on role membership see "Defining Role Membership" on page 55.

Assigning Access

User Domain/Entity Access Maintain provides a workbench type screen for streamlining the setup of access. You can use the three selection criteria fields to limit the records you want to work with or leave them blank to see all combinations in the system. The grid supports standard sorting and group-by features so you can organize the data conveniently.

Clicking in check box in the Select column indicates that the user has access to the associated combination of domain and entity.

Fig. 4.10
User Domain/Entity Access Maintain



You can also modify the user's default domain by selecting the Default Domain check box. Selecting this for one entity in a domain activates the setting for all entities, since the setting applies domain wide.

Note The Primary Entity column is for reference only and cannot be modified here. The domain's primary entity is specified in the Domain Create activity.

Default. Select the check box on the row of the user's default domain. Only one domain can be designated as default. In the .NET UI, the default domain displays only on the first login; on subsequent logins the state of the last session displays.

Note In a multiple-database environment, a user's default domain must be associated with the current database; it cannot be a connection record.

When a user logs in to the database, the system retrieves the information associated with the user's ID. In the character interface, a user with access to more than one domain is prompted for a domain code, which defaults from the record marked as default.

A user with only one assigned domain does not see this prompt at login but is automatically logged in to the single domain associated with the ID specified.

Users employing the .NET UI can switch entities by opening a different workspace.

Setting Up Roles

Roles are used to model the business processes that exist within a business enterprise. Roles determine the set of application resources that display for that user when they access their permitted workspaces. In order to model your organization's business processes effectively, users need access to all the appropriate application resources required for them to perform their everyday business tasks.

In this context, an application resource typically is an executable program that exists within the menu system: either a standard program or a component-based activity. However, in addition to functions executed from the menu, some activities that are not on the menu can be secured.

All system users must be assigned to at least one role in order to gain access to the system. Typically the same role is given to more than one user in an organization, and a single user may have several assigned roles.

Note A user assigned to multiple roles has access to the combination of resources defined in the roles.

Role-based access control provides flexibility and consistency in the way security requirements are enforced, and also helps reduce maintenance for the system administrator. While your users may change based on terminations or task reassignments, roles within an organization typically remain stable over time.

Roles are not domain specific—they are defined system wide. However, roles operate within the context of the domains and entities to which the user has been granted access. This concept is known as *role membership*. See "Defining Role Membership" on page 55.

Uses of Roles

The primary use of roles is to limit access to menu-level functions. Roles are also used to:

- Limit access to other resources such as sites and GL accounts. This is described in Chapter 5, "Setting Up Additional Types of Security," on page 59.
- Limit access to a set of activities that are not on the menu related to component-based functions.
- Create customized versions of component functions that are stored and retrieved at the role level.
- Create saved browse settings and report variants that are stored and retrieved at the role level.

The last two activities are described in *User Guide: QAD Financials*.

Note The Process Maps display on the menu in the .NET UI, but are not secured through role permissions. Anyone can view the maps. However, security is invoked when a user clicks a link in a process map that executes a menu-level program. If they do not have access, an error displays.

Default Roles

Each user can be assigned a default role in Role Membership Maintain. This default is not related to security. For security, a user always is granted the sum of resources assigned to the various roles assigned to them. However, for customizations, searches, and report variants saved at the role level, a default role is required to determine what to display.

Example Customized versions of Supplier Invoice Create are developed for role SalesClerk and SalesManager. The operations manager is assigned both of these roles, but SalesManager is marked as the default role. When the operations manager uses Supplier Invoice Create, the version customized for SalesManager displays.

If a user is not assigned a default role when multiple role-specific customizations exist, the system-level version of the function or report displays.

Non-Menu Resources

Most resources assigned to a role represent menu-level programs and activities. However, roles can be granted permission to a few system activities that are not on the menu.

Table 4.1 shows the activities that must be assigned permissions, but which do not appear on the application menu.

Table 4.1
Secured Items Not on Menu

Secured Item	Description
Customization – Design Mode General Customization – Design Mode Role Customization – Design Mode User	Determines if this role can customize the user interface through the Design Mode features at the system, role, or user level. See the section on design mode in *User Guide: QAD Financials* for details.
Supplier – Supplier Invoices (Entity)	Determines if this role can access the Supplier Invoices (for the current entity) Related View as a right-click option on Supplier browses.
Customer – Customer Invoice (Entity)	Determines if this role can access the Customer Invoices (for the current entity) Related View as a right-click option on Customer browses.
Customer – Customer Invoices Activity	Determines if this role can access the Customer Invoices Activity Related View as a right-click option on Customer browses.
Journal Entry – Create (External)	Determines if this role can create journal entries using an API. The API create method is used by both Operational Transaction Post (25.13.7) and Invoice Post and Print (7.13.4) to create journal entries. It could also be used to post transactions from an external system. You must assign this resource to any users that will be posting operational transactions to the GL.
Posting - Create External Posting (Entity)	Determines if this role can post transactions to external systems from the current entity during Operational Transaction Post (25.13.7).
Stored Search Maintain on Role Level Stored Search Maintain on System Level Stored Search Maintain on User Level	Determines if this role can save stored searches at the role, system, or user level. See *User Guide: QAD Financials* for details.

Secured Item	Description
Tax Code – Create Tax Code – Modify Tax Code – Delete	Determines if this role can create, modify, or delete tax rates with Tax Rate Maintenance (29.4.1). A role must have access to both Tax Rate Maintenance and one of these options to successfully create, modify, or delete tax rates. Tax rates are described in the chapter on Global Tax Management in *User Guide: QAD Global Tax Management*.
User Create User Delete	Determines if this role can create or delete a user with User Maintenance (36.3.1). A role must have access to both User Maintenance and these two options to successfully create or delete a user. See “Defining Users” on page 38.
Report Variant Maintain on Role Level Report Variant Maintain on System Level Report Variant Maintain on User Level	Determines if this role can save report variants at the role, system, or user level. See *User Guide: QAD Financials* for details.
Report Schedule Maintain Report Schedule View	Determines if this role can maintain and view report schedules.

System-Supplied Roles

During system installation, a number of roles are set up automatically. Table 4.2 lists these roles and their function.

Table 4.2
System Roles Created During Installation

Role	Description
_EveryOne	This role is only present in systems that were converted from an earlier version of QAD software. It includes all users that were defined in the previous system.
CustomerNotify	Members of this role receive e-mail notification when a new customer record is created with Customer Create so that the operational data can be completed in Customer Data Maintenance (2.1.1).
EmployeeNotify	Members of this role receive e-mail notification when a new employee record is created with Employee Create so that the employee can be defined as a service/support engineer in Engineer Maintenance (11.13.1).
EndUserNotify	Members of this role receive e-mail notification when a new end user record is created with End User Create so that the operational data can be completed in End User Data Maintenance (11.9.1).
SuperUser	This role provides initial access to all menu functions and is typically assigned to users with an administrative role during system implementation.
SupplierNotify	Members of this role receive e-mail notification when a new supplier record is created with Supplier Create so that the operational data can be completed in Supplier Data Maintenance (2.3.1).

The SuperUser role is initially defined to provide permissions for all menu functions loaded in the system. However, this is true only initially. If you add new menu items manually using Menu System Maintenance (36.4.4.1), you must also manually grant users rights to these menu items in Role Permissions Maintain. When you add new domains and entities, you must explicitly grant access to the SuperUser role for members of this role to continue to have access throughout the system.

Note This is important for certain roles that are used, for example, by daemon processes and require access to all system resources. See “Types of Users” on page 38.

You should define other system roles for special functions such as:

- An administrative role specified in Security Control (36.3.24) to receive e-mail notifications when specific security and controlled events occur.
- The .NET User Interface includes some administrative functions that can be assigned to a specific role.

Role Example

A system administrator configures the system to control access to three functions based on each employee’s organizational level. Three types of access to financial functions are required: one for clerks, one for managers, and one for the CFO.

The system administrator creates three roles: *Clerk*, *Manager*, and *CFO*. Sara, the AP Clerk, is assigned to the Clerk role. Don, the AP Manager, is assigned to the Manager and Clerk roles. Jane, the CFO, is assigned all three roles. In this setup, illustrated in Figure 4.11, Jane’s roles grant her entry to all the levels she is authorized to access.

Fig. 4.11
Using Roles to Give Access

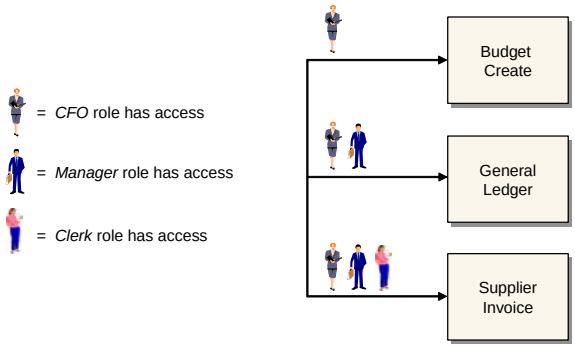


Table 4.3 shows how the system administrator assigns users to each role.

Table 4.3
Sample Role Setup

Role	User
Clerk	Sara, Don, Jane
Manager	Don, Jane
CFO	Jane

Next, the administrator uses Role Permissions Maintain to assign the appropriate system resources to the relevant roles to determine access to the system resources that each user requires in order to complete their assigned tasks.

When Mark is hired as the new deputy CFO, the system administrator only has to assign Mark to the *CFO* role in order to give him access to each individual protected financial function.

When a member of the SalesClerk role logs in, the user has access to:

- Sales Order Maintenance
- Customer View
- Customer Credit View

Instead of seeing the entire set of menus, only Customer Management and Financials display. Within these folders, only the selected functions SalesClerk can access display.

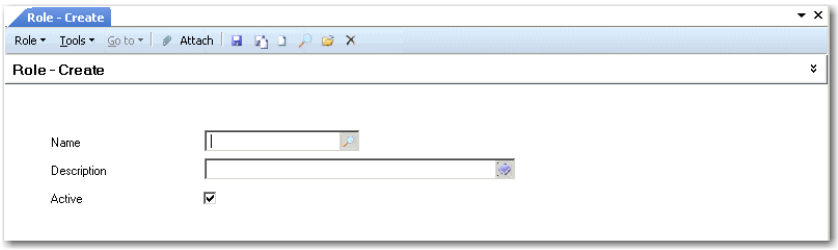
Note Using features of the .NET UI, users can also create their own custom menu display under Favorites.

Defining Roles

Use Role Create (36.3.6.1) to define roles in your system. You should define as many roles as required in order to model your business processes in the system. Use Role Modify (36.3.6.2) to perform maintenance on existing roles defined in your system, and Role View (36.3.6.3) to view roles.

A role defined in the system can be deleted using Role Delete (36.3.6.4) as long as the role is not referenced in the system.

Fig. 4.12
Role Create (36.3.6.1)



Name. Enter a name (maximum 20 characters) identifying a role. Names are restricted to the characters A–Z, a–z, and 0–9.

Note Superuser is an existing role and cannot be used as a name. This role is used during initial system setup and has access to all system functions.

Description. Enter a description (maximum 40 characters) of the role. You can optionally enter descriptions in more than one language. See the section “Using the Translation Option” in *User Guide: QAD Financials*.

Both the role name and description display in the lookup associated with role fields and on various reports and inquiries, as space permits.

Active. Indicate if this is an active record.

Although deactivated roles can still display within browses, a deactivated role cannot be selected from other system functions. If a role is deactivated, existing security records defined within the system that use the role are still valid and remain functional. However, no new security records that reference the deactivated role can be created—new role memberships or role permissions, for example.

Defining Role Permissions

Use Role Permissions Maintain (36.3.6.5) to define the role permissions in your system. You define permissions for both resources on the menu and resources that are not on the menu.

For resources on the menu, note the following:

- Only executables can be secured, not folders. The folder represents a container to help logically organize functions, but no security is associated directly with it. If a user does not have access to any executable programs in a folder, that folder does not display on the menu.
- If an executable appears more than once in the menu tree, the same security always applies to it.

  Example For user convenience, Maser Comment Maintenance (`gpcmmts.p`) appears on the menu as 1.12, 2.1.12, 2.3.12, 2.5.12, and 14.12. You cannot give a role access to menu 1.12 and not give them access to menu 2.1.12. The access is associated with gpcmmts.p, not the menu position. The menu numbers are used only to make it easier to logically group programs.
- Regardless of how a menu resource is accessed, the same security applies. For example, related functions can be accessed from a Go To menu only when the current user has access to the destination function. This is also true of the related views and reports; users must have access to the menu-level program to be able to run it from within another function.
- When a new item is added manually in Menu System Maintenance (36.4.4.1), it is initially inaccessible to all users, and consequently, will not be seen on the menu. You must assign the menu to a role before members of that role can see or use the menu.

  Important You should keep this in mind when customizing the menus; if you simply add the item to the menus and check to see if it appears, you will not be able to see it. You must add it to a role and be logged in as a user with that role to see the new menu.
- Menus are cached in memory during login; you must log out and log in again to see any menu changes. Changes to role permissions do not affect any users currently logged in. Changes take effect the next time they log in. This approach avoids performance degradation from continually checking for security changes.

Note Lookups that let you select a record are not separately secured. Access to a function implies access to all lookups used in the function. When a lookup contains links to drill-down browses, these are only secured if they are on the menu. If a power-browse is not secured by being put on the menu, any one can execute it.

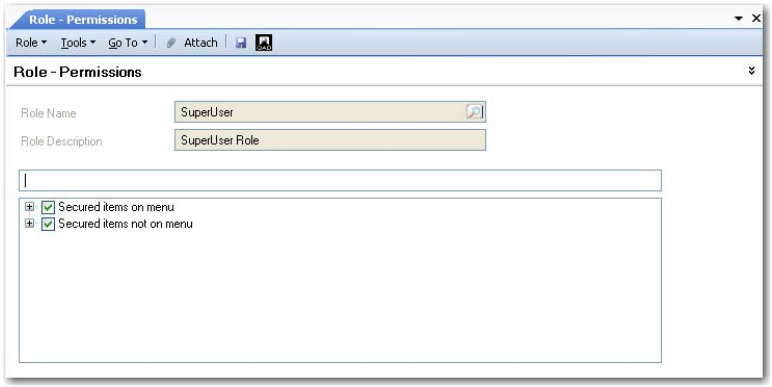
Role-based permission prevents any executable from being run from within your QAD application unless it is added to the menu and then added to a role. You cannot directly invoke a Progress program that is not on the menu by typing its name at the menu prompt in the character user interface.

When users attempt to execute a program on the menu and their current role does not grant access, the message “Program not found” displays.

Important To access the program and field help in the .NET UI, a user must have access to Field Help Maintenance (36.4.13). If a user is unable to access the help, a possible cause of the problem is that the user does not have access to Field Help Maintenance (36.4.13).

Figure 4.13 illustrates Role Permissions Maintain. At the top-most level, you can define secured items on the menu and secured items not on the menu. Both top-level selections display application resources defined in the system using a tree model with leaf and non-leaf nodes, similar to how the menu itself displays.

Fig. 4.13
Role Permissions Maintain (36.3.6.5)

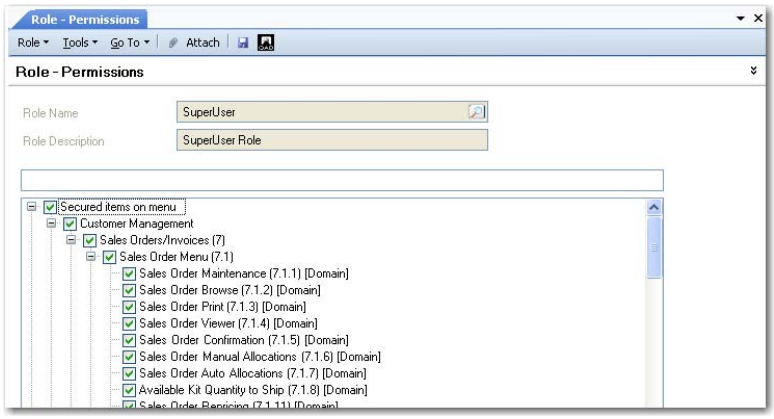


Secured Items on Menu

Each menu item displays identified by menu number and arranged according to the system menu. Any selected items represent the menu items currently assigned to the specified role. Each menu item corresponds to a record in the mnd_det table.

To assign a menu item to a role, open the relevant menu grouping, navigate to the menu item you want to associate with the selected role, and click in the check box next to the item.

Fig. 4.14
Role Permissions Maintain, Secured Items on Menu



The tree nodes can be expanded and contracted. Nodes can have one of three possible states:

- Clear. No selection is in effect.
- Shaded. One or more but not all of the leaf nodes or non-leaf nodes lower in the tree are selected.

- Selected. The top-level node and all lower level nodes are selected.

Selecting a node within a node causes the node higher up to display as shaded. Selecting a node causes all items below it to be selected and is indicated in the higher node. The ability to select a node and cause lower nodes to also be selected streamlines the process of setting permissions.

Since menu items on submenus inherit associations created at a higher level, you can associate a menu group high in the menu tree, clear any unwanted associations for submenus, and then modify selections at the menu item level, if required. This makes the process of administering role permissions easier.

Menu items in the tree display with [Domain] or [Entity] after the menu description. This indicates the level at which security access is checked when this function is executed. Most operational functions are secured at the domain level and most financial functions are secured at the entity level.

- When [Domain] displays, the function can be executed when the user has access to at least one entity within the domain.
- When [Entity] displays, the function can be executed only when the user has access for the specific entity.

Note Do not confuse this with the level of the data being updated by the function. For example, Tax Type Maintenance (29.1.1) is secured at the domain level, even though tax types can be used system wide.

Secured Items Not on Menu

Activities that can be secured even though they are not on the menu are displayed based on the component name and the activities associated with the component. Selected items are currently assigned to the specified role.

You assign items to a role in the same way you assign menu items, by clicking in the check box next to the item.

Fig. 4.15
Role Permissions Maintain, Secured Items Not on Menu

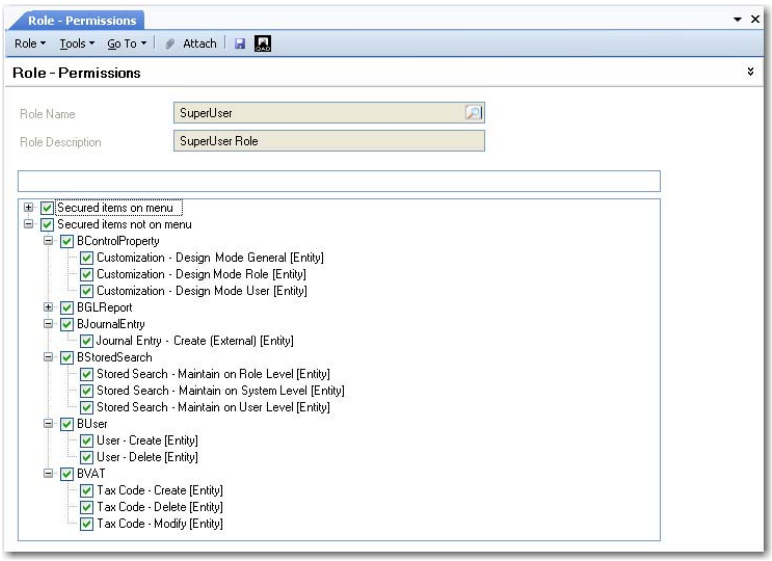


Table 4.1 on page 48 lists these functions and explains their use.

If you plan to let users customize screens for component-based functions to suit their working preferences, you must assign at least one of the Design Mode permissions to the user. If none of these permissions is assigned, the Design Mode menu option does not display on the Tools menu for that user. For details on customizing screens, see the chapter on customization in *User Guide: QAD Financials*.

Defining Role Membership

Use Role Membership Maintain (36.3.6.6) to define an association between a role defined in the system and a system user and to indicate which role is the user's default role. The default role does not affect security, but is used to determine role-specific customizations and stored searches. See "Default Roles" on page 47 for details.

The screen presents a workbench-type interface where you can select records to update by user, role, domain, and entity.

Note Although you can leave all fields blank when generating a list, this is not recommended since the list may take a long time to display, depending on the number of records in the database.

Role membership is always qualified by domain and entity. This is true even though for most operational functions, the specific entity is not relevant. To execute these operational functions, a user must still belong to a role that has access to at least one entity—typically the primary entity—in the domain.

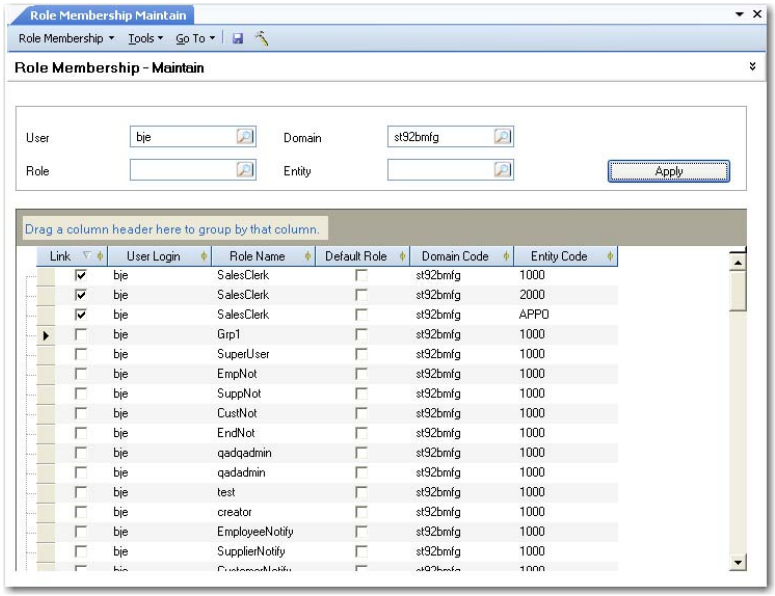
Specifying role membership defines both who belongs to the role and the *role context*; that is, which domains and entities the role can access.

When a user logs in, the system builds the menu for that user by combining:

- All standard functions belonging to all roles assigned to the user in any entity of the selected domain
- All component activities belonging to all roles assigned to the user in the selected entity

Example Domain A has entities located in California, New York, and London. Carol has been assigned the role HR Manager for all three entities. Tom has been assigned the same role, but only for London; Pam and Tom share responsibilities for London. Pam's role membership is specified for entities California, New York, and London. Tom's role membership, however, is defined for the London entity only.

Fig. 4.16
Role Membership Maintain (36.3.6.6)



Use the User, Role, Domain, and Entity fields at the top of the screen to select the records you want to work with during this session. You can group the data in the grid or sort or rearrange columns to streamline the setup activity. Selecting the check box indicates that the user has access to the role for the associated domain and entity.

Note Any changes to a user's domain or entity access privileges also automatically update that user's role membership information. For example, removing a user's ability to access an entity breaks the association between that entity and the user's assigned role, and the entity is deleted from the list of assigned entities in Role Membership Maintain. For details on defining user access to domains and entities, see "Specifying Access to Domains and Entities" on page 45.

Viewing Access Information

The following view programs display security-related information:

- User Domain/Entity Access View displays access privileges that can be filtered on user, domain, and entity.
- Role Permissions View displays permissions filtered on resource or role.
- Role Membership View displays the combinations of role and user filtered by domain, entity, role, or user.
- User Access View is an overall view of all dimensions: user, role, entity, domain, and resource.

Exporting and Importing Roles and Permissions

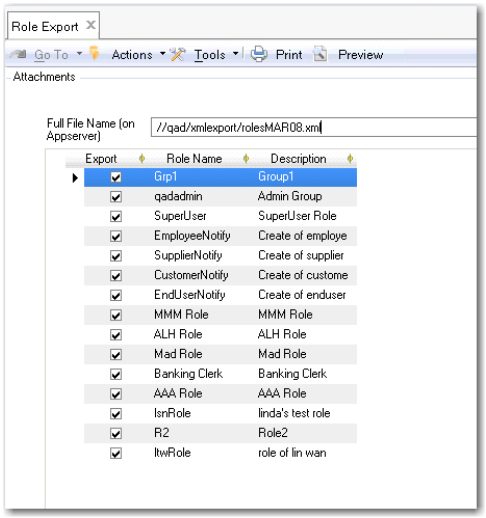
The default roles available in the system are loaded from the file `defaultroles.xml` during the initial install or as part of a system synchronization or update.

Use Role Export (36.3.6.11) and Role Import (36.3.6.12) to export and import roles, their descriptions, and permissions as `.xml` files. The function requires the full file name and location on the application server for the file to be exported or imported. The Import function uses synchronization logic to replace `defaultroles.xml` with the imported file.

The system displays an error message if the file name extension is incorrect, and a confirmation message when the export or import has completed successfully.

Role Export

Fig. 4.17
Role Export



Full Filename (on Appserver). Specify a name and path for the exported file. You must use the XML file extension.

Select the roles to be exported and click Export.

Chapter 5

Setting Up Additional Types of Security

This section discusses how to set up additional types of security for your system.

Additional Security for Component-Based Functions

You can define field security for the component-based functions in the system. This determines whether specific fields are enabled, disabled, or hidden for particular roles in the system.

To use the field security feature, it must be enabled system wide by selecting the appropriate field in System Settings.

Important Enabling field security may result in performance costs to your system. You should take these performance costs into consideration when planning your security implementation in order to reduce unnecessary system load.

Overview of Field Security

Field security lets you indicate that certain fields on a screen either cannot be updated or cannot be seen by users with a particular role. For example, although a user role may have the appropriate permissions to display Customer Invoice Modify, you can prevent users in that role from updating the invoice total field.

Field security controls the ability to modify the UI through the Design Mode features. Fields that are restricted through field security cannot be modified using the UI customization features. Conversely, fields that are not restricted in field security can be customized—including disabling or hiding the field—by using the UI customization features.

Field security can be applied to fields that are delivered with the system as well as user-defined fields.

Field security is defined using any combination of business component, activity, role, and field. A system administrator can specify field access rules that define either rights or limitations:

- Component and field. Define a right or limitation for all users in all activities.
- Component, activity, and field. Define a right or limitation for all users in one activity.
- Component, role, and field. Define a right or limitation for at least one role in all activities.
- Component, role, activity, and field. Define a right or limitation for at least one role in at least one activity.

For each combination, only one rule is possible. The more specific a rule is, the higher its precedence over other rules. For example, component and field is the most general rule; component, role, activity, and field is the most specific rule.

Important In order for any changes to the field security settings to take effect, users must exit from any .NET UI sessions in use while the changes were implemented and then restart the .NET UI.

Setting Up Field Security

In order to use field security, you must:

- Enable field security system wide.
- Assign the Field Security Maintain activity using Role Permissions Maintain.
- Define field security.

Enabling Field Security System Wide

Before field security can be configured, the feature must be activated system wide by using Change System Settings (36.24.5.1) and selecting the Field Security check box.

Important After enabling this setting, you must restart the application servers in order for the change to take effect. The setting change is held in cache until the appserver is restarted, when the change is written to the database.

For more information about the Change System Settings function, see *User Guide: QAD System Administration.*

Assigning the Field Security Activity

Use Role Permissions Maintain (36.3.6.5) to assign the Field Security activity to the appropriate role. On the Role Activities tab, navigate to the Field Security component in the System Administration business grouping and select the Create check box.

For more information on defining role permissions, see “Defining Role Permissions” on page 52.

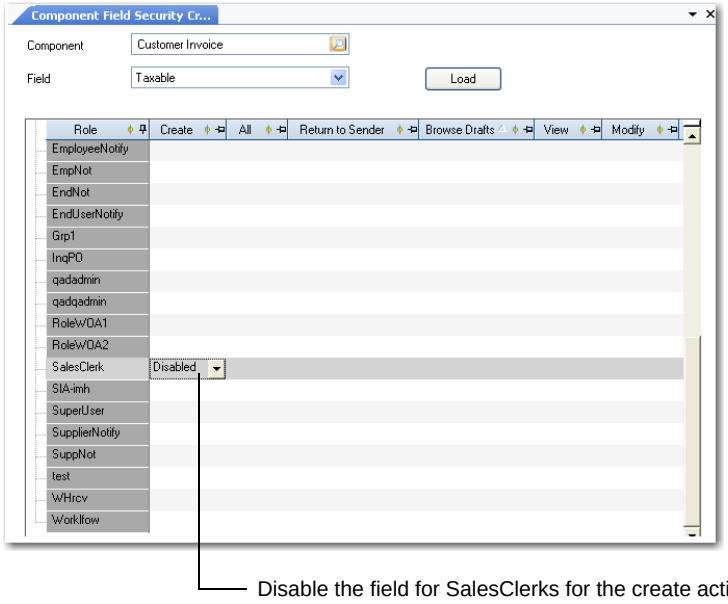
Defining Field Security

Use Component Field Security Create (36.3.15.6) to define field security for user roles.

Select the business component with fields that you want to secure and click Load to display the activities defined for it and the roles that have access rights to the activities. Select an option from the drop-down list at the intersection of a role and an activity to specify the field security for that role/activity combination. Click Save to save your selections to the grid.

Example Select the Taxable field in the Customer Invoice component. In the SalesClerk/Create cell, select Disabled from the drop-down list. This prevents the SalesClerk user role from being able to modify the Taxable setting for new supplier invoices.

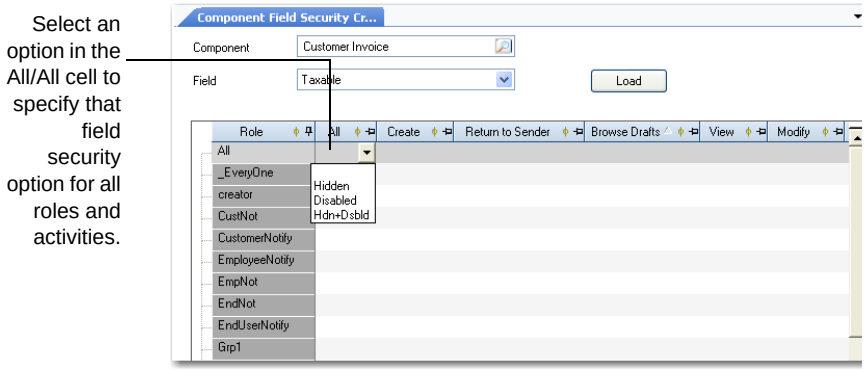
Fig. 5.1
Component Field Security Create (36.3.15.6)



To quickly set field security on the grid, note the following:

- Selecting a field security option on the drop-down list in the All column for an activity specifies that option for all activities for that role.
- Selecting a field security option on the drop-down list in the All/All cell specifies that option for all roles and activities.

Fig. 5.2
Component Field Security Create (36.3.15.6)



To create field security, enter data in the following fields:

Component. Enter the name of a component that contains the field you want to create field security for.

Field. Enter the field you want to define field security for. This menu displays delivered fields as well as user-defined fields.

For a role/activity combination, select an option to set field access rights:

Undetermined: No rule has been defined for this field. User roles have full access rights to this field. This option displays as empty in the drop-down list.

Hidden: The field does not display in the UI. In terms of business logic, any value contained in this field cannot be modified.

Disabled: The field displays in the UI, but any value in the field is locked—data cannot be entered into this field.

Hidden and Disabled: The field does not display in the UI, and cannot be made visible in the UI even if the UI is customized.

Additional Security for Standard Programs

In addition to the standard role-based permissions, a number of other types of security can be defined for standard programs. These types of security apply in operational areas.

Using the system, you can:

- Specify user IDs that can update the value of specific program fields.
- Determine which users or roles can create operational transactions that affect:
  - Sites, locations, inventory statuses, and other inventory-related attributes
  - General ledger accounts
  - Inventory movement codes

Note the following fundamental difference in the way role-based security works and the way these additional forms of operational security work:

- For role-based access to domains and entities or resources secured through Role Permissions Maintain, no one has access unless it is specifically granted.
- For other types of operational security, all users and roles have access unless specific access records have been defined. Once access records have been defined, other users and roles are automatically prevented from having access.

If you use the Sales and Use Tax Interface (SUTI) to communicate tax data between the system and Vertex's Quantum for Sales and Use Tax product, set up similar access controls in Tax Interface Control (36.5.3.24). See *Technical Reference: Sales and Use Tax Interface* for information on SUTI.

Access records apply only to the current domain from which they are entered.

Specifying User IDs and Roles

To define security access by field, site, and so on for standard programs, you can enter any number of valid user IDs and/or roles, separated by commas, in the following programs:

- Specify user IDs in Field Security Maintenance (36.3.15.1). See page 64.
- Specify user IDs or roles in Site Security Maintenance (36.3.13.8). See page 67.
- Specify user IDs or roles in GL Account Security Maintenance (36.3.13.1). See page 78.
- Specify user IDs or roles in Inventory Movement Code Security (36.3.13.13). See page 78.

Note If you do not set up records in these programs, the system by default allows access to all users who pass login, domain, and role-based access security restrictions. See “Login Security” on page 15.

The system validates entries against records set up in User Maintenance and Role Create.

The asterisk (*) and exclamation point (!) are special characters when used in the User IDs/Roles field.

- The asterisk (*) gives access to all users and roles.
- The exclamation point restricts specific users by user ID, not by role. For example, `!user1,*` means all users except user1 have access to the function; `!user1,admin` allows access only to members of the admin role, with the exception of user1. However, `!admin,*` does not prevent members of the admin role from accessing the function.

When using the exclamation point, you must enter exclusions first: `*,!user1` gives access to all users *including* user1. To exclude multiple users, enter:

```
!user1,!user2,!user3,*
```

Important When you enter exclusions, you must also define users who have access. For example, if you enter just `!user1`, you are specifying that `user1` does not have access—but you have not granted access to other users. The result is that no one has access to the controlled function. To avoid this situation, be sure to enter the appropriate user IDs, roles, or an asterisk after the exclusions. In this example, `!user1,*` excludes user1, but lets all other users run the program.

When you use the asterisk to grant access to all but specifically excluded users, the logic works correctly only when excluded users are not assigned to roles. The asterisk allows access to all users assigned the role, even if they have been excluded as individuals.

Table 1.6 lists some examples. User IDs and role names are not case-sensitive.

Table 5.1
Sample Uses of User ID and Role Name

String	Description
*	All users have access.
`mary, manager`	Only user mary and members of the manager role have access.
`!jcd,*`	Everyone but user jcd has access.

The inverse of the last example does not work. If you put `*,!jcd` in the field, the system grants everyone access first and does not go back to check on `jcd`. Someone using the `jcd` user ID would not be excluded. In general, avoid using any exclamation point after the very beginning of the entry.

Limiting Access to Fields

Field security prevents unauthorized users from updating secured fields in standard programs. Field security does not prevent users from seeing the value of a field if they have access to the screen where it is updated. Nor does it protect a field from program-level updates through custom code.

Note Standard field security is not exactly the same as component-based field security. In component-based functions, you can disable and hide fields.

The system determines whether a user is authorized based on whether the user ID matches the values specified for the field.

Field Security Validation

When you install your QAD application, security is not active for any fields, and only a few fields are eligible for field security. Use the Dictionary Field Security Report (36.3.15.4) to determine which fields can be given security.

In the character interface, you also can access the field on a screen and press Ctrl+F. The information window indicates whether password validation is available for the field.

An eligible field must have a specific validation expression in the data dictionary that references `gppswd.v`. The syntax is:

```
{gppswd.v &field=<dictionary field name>}
```

Activated Field Security Report

Use the Activated Field Security Report (36.3.15.3) to see which fields have security activated. It also lists privileged user IDs.

Dictionary Field Security Report

The Dictionary Field Security Report (36.3.15.4) lists the fields containing the association to the validation file as part of their definition.

Protect any of these fields from update by creating a record of privileged user IDs or roles. This association can be made to any field, and is one of the only database definition changes you can make that does not constitute a schema change.

Adding Security to an Eligible Field

1. Add the field name and the list of user IDs that can access the field in Field Security Maintenance (36.3.15.1).
2. Verify that the field is secured by running the Activated Field Security Report (36.3.15.3).

Adding Field Security Eligibility

You can make most fields eligible for field security by adding the validation expression to the field in the data dictionary. You then recompile the programs that use the field, using the modified data dictionary. It is not always possible to add field security. Some fields have preexisting data dictionary validation expressions that prevent the addition of `gppswd.v`.

Warning Once you have made a field eligible for field security, you cannot make it ineligible. You can deactivate the security by removing all user IDs for the field in Field Security Maintenance (36.3.15.1).

For multiple databases, make your security changes in the database against which you compile. The changes are then in effect for any other databases you run the compiled code against.

1. Identify and list all fields to which you want to add security.

   Since recompiles take time, it is more efficient to add all field security at once.

2. Make sure all other users are logged out.

3. Run Field Eligibility Maintenance (`mgfldcmt.p`, 36.25.22), which changes the validation expression and message in the data dictionary.

4. Set field security for each field on your list.

   The `mgfldcmt.p` utility prompts for a table and field name on which to activate field security. Once you enter a valid field and table name and you press Next, you are prompted for the next entry.

5. Press End to exit Field Eligibility Maintenance.

6. Recompile either all programs or those programs impacted by the changed field security. If you have custom programs that access these fields, they also need to be recompiled.

   To compile only the affected programs, make a backup copy of `utcompil.wrk` in the `qad` directory, and then delete the program names that you do not want recompiled from the file.

   `utcompil.wrk` contains a complete list of all programs.

7. Back up recompiled code.

8. You can now add the field name and the list of user IDs that can access each field in Field Security Maintenance.

9. Verify that each field is secured by running the Activated Field Security Report.

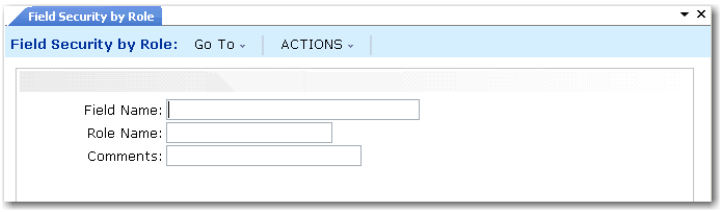
Field Security by Role

You also can set up field security for all users that are assigned to a specific role.

1. Assign users to roles in Role Membership Maintain (36.3.6.6.1).

2. Execute Field Security by Role (36.3.15.2). This function adds all users who belong to a specified role or roles to the list of authorized users for a validated field.

Fig. 5.3
Field Security by Role (36.3.15.2)



Even with this process, field security is only available at the user level, not the role level. Field Security by Role is simply a batch utility that lets you add multiple users simultaneously. This has the following consequences:

- If you remove a user from a role that was given access to a field, that user can still access the field. To prevent this, use Field Security Maintenance (36.3.15.1) to remove the individual user.
- You cannot use Field Security by Role to remove multiple users from the list of authorized users. To remove multiple users, you must remove users individually in Field Security Maintenance.
- If you delete a role in Role Delete (36.3.6.4), individual records remain on the system until you delete them in Field Security Maintenance.

Once Field Security by Role is executed for a field and role, all users who belong to the role display in Field Security Maintenance as authorized to access the field. The Comments field in Field Security by Role displays as the comment for the field and user combination in Field Security Maintenance.

Controlling Inventory Updates

The system has two ways to control inventory updates within a domain. Updates can be controlled by:

- Sites within a domain
- Specific combinations of inventory-related fields such as item number and location

Note If you are using the optional QAD Warehousing module, you can also assign security by warehouse using Warehouse Security Maintenance (4.23.13).

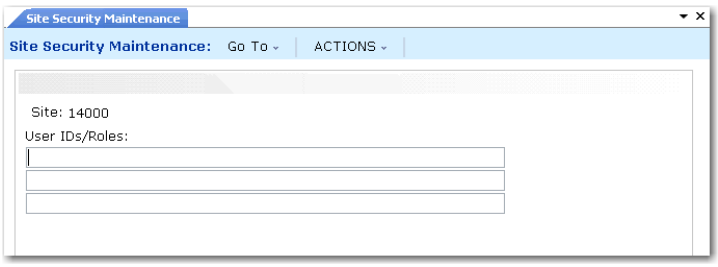
Access by Site

Site security lets administrators control user access to inventory transactions at each site in a domain. Only authorized users can process transactions at secured sites.

Note By default, users have access to all sites unless security has been defined in this program.

Access is managed by user and role. A user can access a site only if that user's ID or role is specified in the User IDs/Roles field in Site Security Maintenance (36.3.13.8). Use Site Security Report (36.3.13.9) to view site security defined for system users.

Fig. 5.4
Site Security Maintenance (36.3.13.8)



When a user enters a restricted site code in a site-controlled program, the system checks the value of the User IDs/Roles field associated with the site in Site Security Maintenance. If the user does not belong to an appropriate role, or if the user is not given specific access by user ID, an error message displays and the user cannot complete the transaction.

Programs Affected

- Site security works with programs that change inventory data and have a Site field as part of the selection criteria.
- Site security checks ranges of sites on batch update programs that meet the previous criteria: they affect inventory and have a Site field. This includes programs such as Regenerate Materials Plan (23.2) and Sales Order Auto Allocations (7.1.17).
- Site security does not affect inquiry and report programs.
- Delete and archive programs, Contract Control (11.5.24), and Quality Management Control (19.24) do not use site security.
- You must set up each domain individually.

Implementing Site Security

It is important to plan site security carefully and to follow closely the procedures for creating roles, users, and associations between users and roles. After you have created any site security records, users who are not listed individually or who have not been assigned access privileges in Site Security Maintenance (36.3.13.8) cannot complete transactions at secured sites.

Ranges of Sites

Many programs let you access a range of sites at one time. Site security controls data updates and processes for ranges of sites. If you enter a range of sites, you must have access to all of them for the update to occur.

When you enter a range of sites that includes sites you do not have access to, an error message displays for the first site code from which you are restricted. You must then adjust the site range to include only sites that you can access.

Update Restrictions

You can use Site Security Maintenance to specify which roles are allowed to update inventory at particular sites within a domain. This type of security is discussed in the section “Access by Site” on page 67.

However, when stringent internal controls for regulatory reporting exist, you may need to control who can update inventory at a more detailed level than the site. For example, you might need to allow only certain roles to transfer inventory out of a Quarantine location, or restrict certain roles from making specific inventory status code changes within a site.

You can use the programs on the Update Restrictions Menu to implement stricter control over inventory movements and status updates that are completed throughout the system. Each particular update restriction program affects a set of programs where inventory transactions occur. Grouping the transactions this way lets you focus control on the areas that are critical to your business practices.

The specific combination of fields that you use to manage inventory depends on the particular program that supports the update restriction, but can include item number, site, location, inventory status code, and the GL account affected by the transaction.

Using the features of Update Restrictions, you can:

- Authorize a role to maintain records, create transactions, or change inventory status for specific item, site, and location combinations in a set of programs that create transactions.
- Restrict a role from maintaining records, creating transactions, or changing inventory status for specific item, site, and location combinations in certain system programs.
- Generate a report of all defined update restrictions.

You should define update restrictions as part of a general security model, which includes defining roles for your organization, as well as setting up any required site and field security. The system applies site and field security before update restrictions.

Note The descriptions of the programs in this section only discuss fields that are unique to that program. The item, number, site, and location fields operate identically in each program.

Setting Up Update Restrictions

You can set up update restrictions for the following types of functions. “General Rules for Update Restrictions” on page 70 describes the common logic used in all of the functions to determine whether restrictions apply to a user’s role.

- Inventory transfers. See “Inventory Transfer Restriction Maintenance” on page 71 for details.
- Inventory details. See “Inventory Detail Restriction Maintenance” on page 72 for details.
- Issues and receipts. See “Unplanned Issue/Receipt Restriction Maintenance” on page 73 for details.
- Maintaining all types of purchase orders. See “PO Restriction Maintenance” on page 74 for details.
- Receipt handling and status change. See “PO Receipts Restriction Maintenance” on page 74 for details.
- Maintaining sales order, invoice, and quote lines. See “SO Restriction Maintenance” on page 75 for details.
- Creating shipments and maintaining shippers. See “SO Shipments Restriction Maintenance” on page 75 for details.
- Maintaining distribution orders (DO). See “DO Restriction Maintenance” on page 76 for details.
- Creating DO shipments. See “DO Shipments Restriction Maintenance” on page 76 for details.
- Creating DO receipts. See “DO Receipts Restriction Maintenance” on page 77 for details.
- Maintaining records, creating receipts, and completing shipments in SSM. See “SSM Restriction Maintenance” on page 77 for details.

Update restrictions only apply to the domain where they are defined. If no restrictions are defined, all roles with access to these programs can perform data updates, as long as they also have access to the site being updated.

Using Wild Cards for Update Restrictions

You can enter item number, site, location, and status values using wild cards. Use the asterisk (*) and exclamation point (!) as inclusion and exclusion wild cards respectively:

- * indicates access to all.

- string* indicates access to all beginning with string.
- string indicates access to that string only.
- !string* indicates access is restricted from all beginning with that string.
- !string indicates access is restricted from that string only.

Use combinations of inclusive and exclusive restrictions to specify both large and small sets of value combinations.

General Rules for Update Restrictions

1. If one restriction is set up, all transactions are validated and have to pass to be accepted. If no restrictions are set up, all transactions are accepted.

   Example As soon as a restriction is set up in Inventory Restriction Detail Maint (36.3.7.2), then Inventory Detail Maintenance (3.1.1) is secured for all transactions.

2. Defining a record in one of the restriction setup programs only affects the programs covered by this restriction. For example, defining a restriction in Inventory Restriction Detail Maint has no effect on Sales Order Maintenance because Sales Order Maintenance restrictions are specified in SO Restrictions Maintenance (36.3.7.8).

3. The values from the transaction are matched against the restrictions defined. There has to be one positive match for the transaction to be accepted.

4. If there is more than one match, all must be positive matches for the transaction to be accepted. In other words, the first negative match will make the transaction fail.

5. There is an implied hierarchy within the fields used to set up the restrictions (Item, Site, Location, and so on). The hierarchy is the order in which they appear on the screen where the restrictions are set up. If a transaction matches a field value that is excluded for a role—for example, Site: !10000—the system fails the transaction without considering the values of the fields lower in the setup screen.

6. These rules are the same for all transactions that are enabled for this functionality. The fields may be different from one transaction to another, but the above rules still apply.

Examples of Update Restrictions

The examples in this section relate to Inventory Transfer Restriction Maintenance. However, as the programs on the Update Restrictions Menu all operate in a similar manner, these examples are relevant to the other Update Restrictions programs.

Example For the specified role, one restriction is defined for a combination of values:

Role (Stock), Item (22-100), From Site (10000), To Site (11000), From Location (100), To Location (200), From Status (20), To Status (20)

Only users with the Stock role can create transactions for this combination of values. This role cannot create transactions for any other combinations; no other role can create transactions for any combinations at all.

Example For the specified role, restrictions are defined for multiple items:

Item (22-1*), From and To Site (10000), From and To Location (100), From and To Status (*)

Item (22-2*), From and To Site (11000), From Location (100), To Location (200), From and To Status (*)

With these restrictions defined, only the specified role can create transactions for items with any status where:

From and To Site is 10000, From and To Location is 100, and item numbers begin with 22-1

From and To Site is 11000, From Location is 100, To Location is 200, and item numbers beginning with 22-2

No other role can create transactions for any combinations at all.

Example When users will be prevented from creating transactions for items with only a few combinations of values, it is best to first define restrictions that allow all changes; then define additional restrictions for the small set of value combinations where changes are prevented.

For each role, set access to all (*) for all items, sites, locations, and statuses; then set the further restrictions:

Item (*), From Site (*), To Site (*), From Location (*), To Location (*), From Status (*), To Status (*)

Item (!22-200*), From Site (10000), To Site (10000), From Location (100), To Location (200), From Status (*), To Status (*)

For each role where these restrictions are defined, they prevent transfers from location 100 to 200 for items that begin with 22-200 in site 10000. Transfers are still allowed for items with all other value combinations.

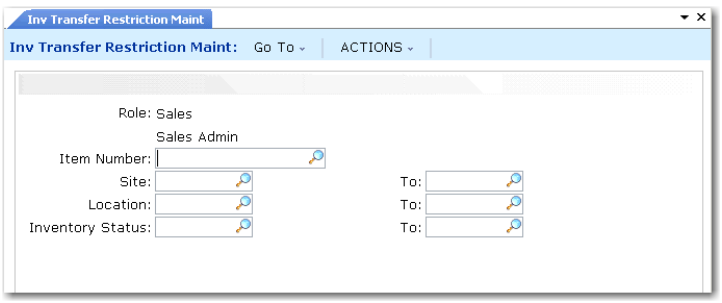
Inventory Transfer Restriction Maintenance

Use Inventory Transfer Restriction Maintenance (36.3.7.1) to specify role-based restrictions on inventory transfers in the following programs:

- Transfer – Single Item (3.4.1)
- Transfer – Multi Item (3.4.2)
- Transfer with Lot/Serial Change (3.4.3)
- Batchload Transfer with Lot/Serial Change (3.4.4)

For each item, restrictions are based on combinations of values for From and To sites, locations, and inventory statuses. All fields are required.

Fig. 5.5
Inventory Transfer Restriction Maintenance (36.3.7.1)



Role. Enter the name of a role defined in your system.

Use roles to streamline security setup. When a new user record is created, you assign the user a role to ensure they have correct access.

Item Number. Enter the code identifying an inventory item defined in Item Master Maintenance (1.4.1).

Item codes uniquely identify items or products. These may be raw materials, purchased or manufactured intermediates, finished items, or packaging materials. Item codes are also used to identify planning items, configured products, repair parts, service items, and kits.

Most reports and inquiries can be selected by item number.

Site. Enter the originating site that is part of the definition for this update restriction.

To. Enter the receipt or receiving site that is part of the definition for this update restriction.

Location. Enter the originating location that is part of the definition for this update restriction.

To. Enter the receipt or receiving location that is part of the definition for this update restriction.

Inventory Status. Enter the original or starting status code that is part of the definition for this inventory restriction.

Restrictions can specify whether this status code can be changed to another specified status code or whether it can be changed at all. This is part of a restriction definition that specifies which roles have access to update specific combinations of items, sites, and locations.

Define status codes in Inventory Status Code Maintenance (1.1.1). Assign inventory status codes to sites with Site Maintenance (1.1.13) and locations with Location Maintenance (1.1.18). Optionally assign default inventory status codes for purchase order or work order receipts to individual items using Item Master Maintenance (1.4.1), Item Inventory Data Maintenance (1.4.5), or Item-Site Inventory Data Maintenance (1.4.16).

To. Enter the destination or target status code that is part of the definition for this inventory restriction.

Inventory Detail Restriction Maintenance

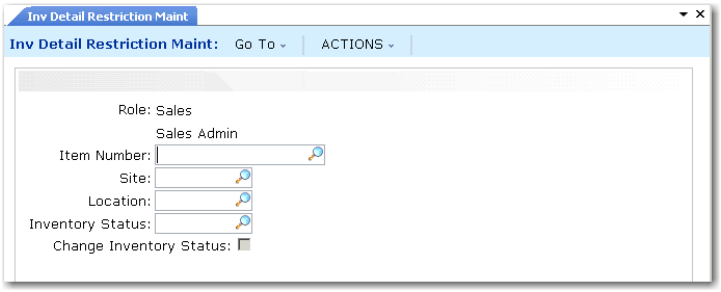
Use Inventory Detail Restriction Maintenance (36.3.7.2) to specify role-based restrictions on updating inventory details in these programs:

- Inventory Detail Maintenance (3.1.1)
- Detail Maintenance by Item/Lot (3.1.2)

Multiple To statuses can be specified for each site, location, and From status combination for an item. Select the Change Inventory Status check box and press Next; this displays the Valid Inventory Status Code frames for maintaining the list of valid To statuses. Enter a status code in the bottom frame and press Next to add it to the list of valid codes.

If the Change Inventory Status is not selected, the system deletes the list of valid To status codes defined for the current role, item, and values combination.

Fig. 5.6
Inventory Detail Restriction Maintenance (36.3.7.2)



Change Inventory Status. Indicate how you want this update restriction to apply to inventory status changes:

Not selected: Users can change other inventory details such as assay percentage and expire date but they cannot change the inventory status.

Selected: Users can modify the inventory status associated with records for the specified combination of item, site, and location as long as they select an inventory status associated with this restriction.

Selecting the check box displays the list of target status codes that are allowed and lets you add or remove codes as required.

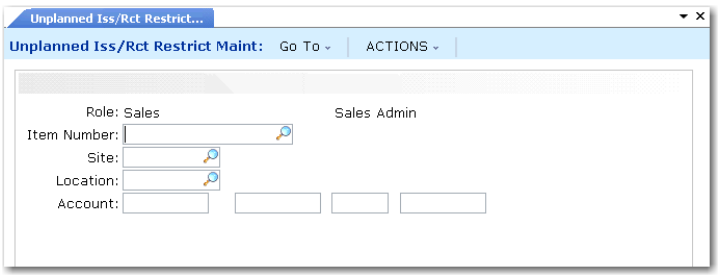
Status Code. Enter a status code to add to the list of valid target status codes that the current code can be changed to. The list is part of an update restriction defined for a particular role and combination of item, site, and location.

Unplanned Issue/Receipt Restriction Maintenance

Use Unplanned Issue/Receipt Restriction Maintenance (36.3.7.3) to define role-based restrictions on issues and receipts in these programs:

- Issues–Unplanned (3.7)
- Receipts–Unplanned (3.9)
- Receipts–Sales Order Return (3.10)
- Receipts–Return to Stock (3.11)
- Receipts–Backward Exploded (3.12)

Fig. 5.7
Unplanned Issue/Receipt Restriction Maintenance (36.3.7.3)



Account. Enter a GL account code as part of an issues and receipts update restriction defined for a particular role and combination of item, site, and location.

The account number is used when applying update restrictions to programs such as Issues–Unplanned (3.7) and Receipts–Unplanned (3.9). Define account codes in Account Create (25.3.13.1).

Sub-Account. Enter a GL sub-account code as part of an issues and receipts update restriction defined for a particular role and combination of item, site, and location.

The account number is used when applying update restrictions to programs such as Issues–Unplanned (3.7) and Receipts–Unplanned (3.9). Define sub-accounts in Sub-Account Create (25.3.17.1).

Cost Center. Enter a cost center code as part of an issues and receipts update restriction defined for a particular role and combination of item, site, and location.

The account number is used when applying update restrictions to programs such as Issues–Unplanned (3.7) and Receipts–Unplanned (3.9). Define cost centers in Cost Center Create (25.3.20.1).

Project. Enter a project code as part of an issues and receipts update restriction defined for a particular role and combination of item, site, and location.

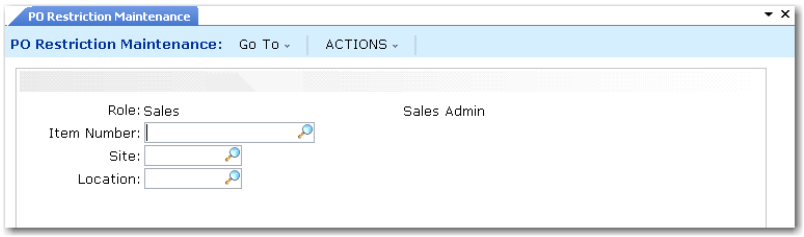
The account number is used when applying update restrictions to programs such as Issues–Unplanned (3.7) and Receipts–Unplanned (3.9). Define projects in Project Create (25.3.11.1.1).

PO Restriction Maintenance

Use PO Restriction Maintenance (36.3.7.5) to specify role-based restrictions on maintaining orders in these programs:

- Build PO from Requisitions (5.2.18)
- Blanket Order Maintenance (5.3.1)
- Blanket Order Release to PO (5.3.6)
- Scheduled Order Maintenance (5.5.1.13)
- Purchase Order Maintenance (5.7)

Fig. 5.8
PO Restriction Maintenance (36.3.7.5)



PO Receipts Restriction Maintenance

Use PO Receipts Restriction Maintenance (36.3.7.6) to specify role-based receipt handling and status change restrictions in these programs:

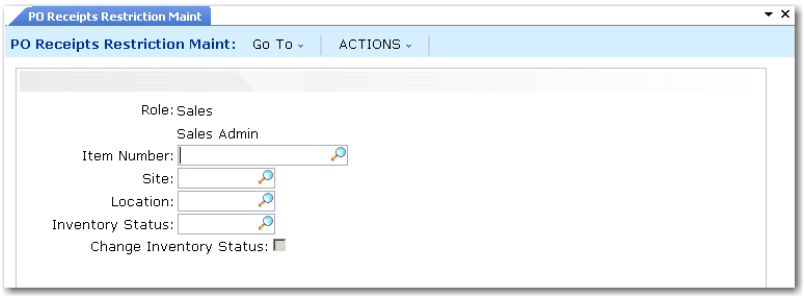
- Purchase Order Receipts (5.13.1)
- Purchase Order Returns (5.13.7)
- PO Shipper Maintenance (5.13.14)

- PO Fiscal Receiving (5.13.16)
- PO Shipper Receipt (5.13.20)

Multiple To statuses can be specified for each site, location, and From status combination for an item. Select the Change Inventory Status check box and press Next; this displays the Valid Inventory Status Code frames for maintaining the valid To status codes. Enter a status code in the bottom frame and press Next to add it to the list of valid codes.

If the Change Inventory Status check box is not selected, the system deletes the list of valid To status codes defined for the current role, item, and value combination.

Fig. 5.9
PO Receipts Restriction Maintenance (36.3.7.6)

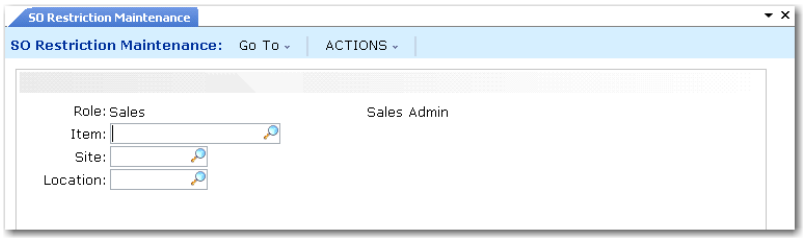


SO Restriction Maintenance

Use SO Restriction Maintenance (36.3.7.8) to specify role-based restrictions on maintaining sales order, invoice, and quote lines in these programs:

- SO Maintenance (7.1.1)
- Scheduled Order Maintenance (7.3.13)
- Pending Invoice Maintenance (7.13.1)
- Sales Quote Maintenance (7.12.1)
- Sales Quote Copy from Order (7.12.5)
- Sales Quote Copy from Quote (7.12.6)
- Sales Quote Release to Order (7.12.10)

Fig. 5.10
SO Restriction Maintenance (36.3.7.8)



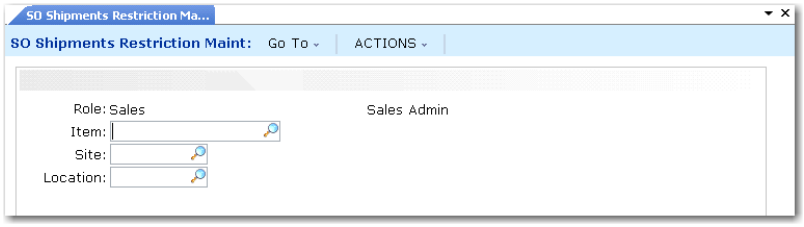
SO Shipments Restriction Maintenance

Use SO Shipments Restriction Maintenance (36.3.7.9) to specify role-based restrictions on creating shipments and maintaining shippers in these programs:

- Picklist/Pre-Shipper – Automatic (7.9.1)

- Pre-Shipper/Shipper Workbench (7.9.2)
- Pre-Shipper/Shipper Confirm (7.9.5)
- Pre-Shipper/Shipper Auto Confirm (7.9.7)
- Sales Order Shipper Maintenance (7.9.8)
- Sales Order Shipments (7.9.15)
- Shipper Unconfirm (7.9.21)

Fig. 5.11
SO Shipments Restriction Maintenance (36.3.7.9)

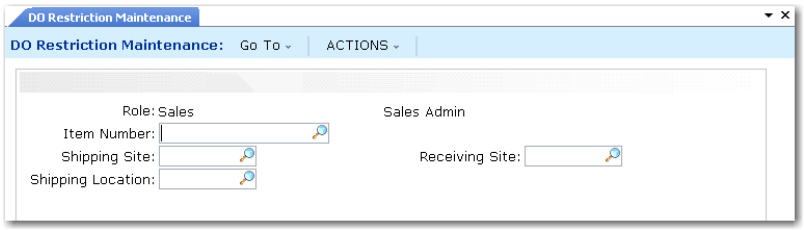


DO Restriction Maintenance

Use DO Restriction Maintenance (36.3.7.13) to specify role-based restrictions on maintaining distribution orders (DOs) in these programs:

- Distribution Order Workbench (12.17.13)
- Distribution Order Maintenance (12.17.14)
- Distribution Order Processing (12.17.21)

Fig. 5.12
DO Restriction Maintenance (36.3.7.13)

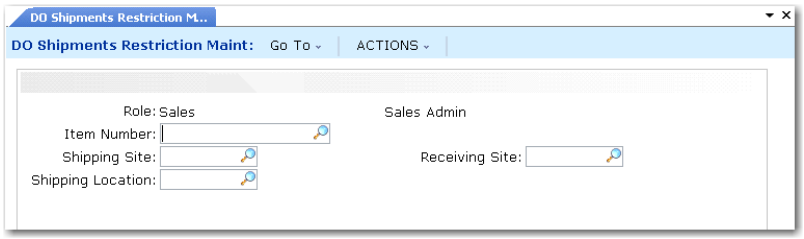


DO Shipments Restriction Maintenance

Use DO Shipments Restrictions Maintenance (36.3.7.14) to specify role-based restrictions on creating shipments in these programs:

- Distributed Order Processing (12.17.21)
- Distributed Order Shipments (12.17.22)

Fig. 5.13
DO Shipments Restriction Maintenance (36.3.7.14)

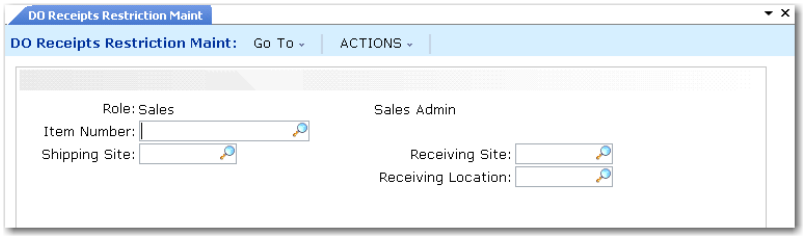


DO Receipts Restriction Maintenance

Use DO Receipts Restriction Maintenance (36.3.7.15) to specify role-based restrictions on creating DO receipts in these programs:

- Intersite Request Maintenance (12.15.1)
- Distributed Order Receipt (12.15.20)

Fig. 5.14
DO Receipts Restriction Maintenance (36.3.7.15)

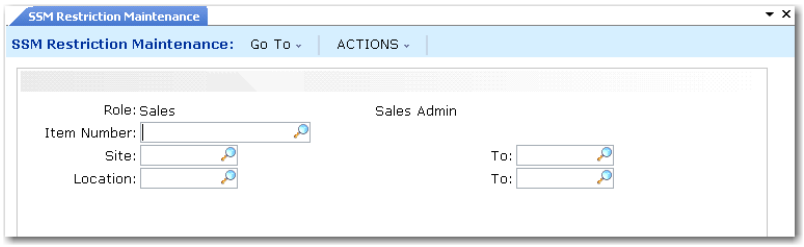


SSM Restriction Maintenance

Use SSM Restriction Maintenance (36.3.7.17) to specify role-based restrictions on maintaining records and creating receipts and shipments for RMAs, RTSs, and MOs in these programs:

- RMA Maintenance (11.7.1.1)
- RMA Receipts (11.7.1.13)
- RMA Shipments (11.7.1.16)
- RTS Maintenance (11.7.3.1)
- RTS Receipts (11.7.3.13)
- RTS Shipments (11.7.3.16)
- Material Order Maintenance (11.11.1)
- Material Order Shipments (11.11.6)
- MO Direct/Pending Returns (11.11.8)

Fig. 5.15
SSM Restriction Maintenance (36.3.7.17)

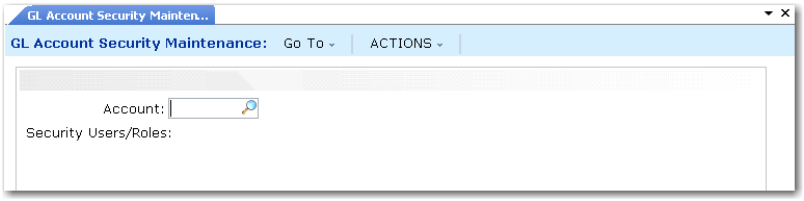


Defining GL Account Security

Using GL account security, you can restrict who can create transactions in operational functions that update GL accounts based on user ID or role.

Use GL Account Security Maintenance (36.3.13.1) to assign users or roles to account numbers. Use the GL Account Security Report (36.3.13.2) to list all accounts that have controlled access.

Fig. 5.16
GL Account Security Maintenance (36.3.13.1)



When a user attempts to create an operational transaction affecting an account, the system checks to see if account security is defined. If it is, the system verifies that the user ID and roles associated with the user are found on the list associated with the account. If a match is not found, a message displays and the user cannot complete the transaction.

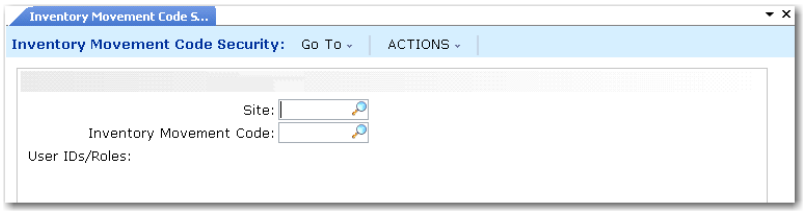
If no account security has been defined, all users and roles are permitted to update the account, within the parameters of other restrictions.

Note Account security is not applied during Operational Transaction Post (25.13.7) or Invoice Post and Print (7.13.4). Use Role Permissions Maintain to restrict posting functions.

Defining Inventory Movement Code Security

Use Inventory Movement Code Security (36.3.13.13) to grant or deny access to individuals and roles to shipping transactions that reference a specific inventory movement code at a particular site.

Fig. 5.17
Inventory Movement Code Security (36.3.13.13)



When you create shippers, the system determines which inventory movement codes are available based on the Ship-From site of the shipper. Access to the inventory movement code also determines if you can select an existing shipper for maintenance. See *User Guide: QAD Sales*.

Note Inventory movement security does not affect whether a line item from a given sales order or other originating transaction can be added to a shipper.

You can delete inventory movement security records at any time.

Use Inventory Movement Code Security Browse (36.3.13.14) to display inventory movement code security records. You can view fields associated with a record by scrolling the display to the left or right. Fields available as filtering parameters in Browse Options are also available on the Sort By selection list.

Chapter 6

Using Electronic Signatures

This section discusses how to set up and use electronic signatures functionality in your system.

Overview

Particularly in areas with critical processes that rely on tight quality control such as the pharmaceuticals industry, regulatory guidance often requires records to be signed by an author, approver, tester, or other accountable individual.

While this signature process is historically associated with a hard-copy signature on paper, it has been extended in many areas to electronic records. For example, the United States Food and Drug Administration (FDA), in 21 CFR Part 11, describes how electronic signatures can be used to support automated processing.

The electronic signatures features of the Enhanced Controls menu support this requirement. You can configure your system to require users of some programs to enter a valid user ID and password before they can create or update records. Additionally, they must provide a reason code that defines the meaning of the signature; for example, Approved or Tested. Based on setup data, users may be able to enter a related remark as part of the signature.

Note Any valid user who has access to a function that records signatures can sign records. Use Role Permissions Maintain (36.3.6.5) to assign access to signature-controlled functions based on user roles. See “Defining Role Permissions” on page 52.

These features are intended as part of an overall approach—also incorporating capabilities offered by system security—to meeting the user accountability requirements of customers with regulated environments.

Eligible Programs

Electronic signature functionality is limited to a subset of programs, tables, and fields that are defined in QAD-provided default signature profiles. See “Profiles” on page 88. Table 6.1 lists the programs that currently can have electronic signatures enabled.

Table 6.1
Programs Included in Default Profiles

Module	Menu	Program
Product Change Control (PCC)	1.9.2.8	PCR/PCO Detail Inquiry
	1.9.6.1	PCR/PCO Approval
	1.9.6.13	Detail Approval Maintenance
	1.9.7.4	Incorporation Selection
	1.9.7.5	Incorporation
	1.9.7.13	Implementation
	1.9.9.1	Print PCR/PCO
Regulatory Attributes	1.22.1	Lot Master Maintenance
	1.22.2	Lot Master Inquiry
	1.22.24	Regulatory Attributes Control

Module	Menu	Program
Inventory Control	3.1.1	Inventory Detail Maintenance
	3.1.2	Detail Maintenance by Item/Lot
	3.4.1	Transfer–Single Item
	3.4.3	Transfer With Lot/Serial Change
	3.4.4	Batchload Transfer with Lot/Serial Change
	3.6.5	Inventory Detail Report
	3.21.1	Transactions Detail Inquiry
	3.24	Inventory Control
Process	4.8.16	Inventory Detail Maintenance
Shop Floor Control	16.20.1	Labor Feedback by Work Order
	16.20.2	Labor Feedback by Employee
	16.20.3	Labor Feedback by Work Center
	16.20.4	Non-Productive Labor Feedback
	16.20.5	Operation Complete Transaction
	16.20.6	Operation Move Transaction
	16.20.13.9	Operation Transaction Detail Inq
	16.20.13.14	Operations By Work Order Report
	16.20.13.15	Operations By Employee Report
Quality Management	19.11	Quality Order Results Entry
	19.12	Quality Order Results Report
	19.13	Test Results Maintenance
	19.15	Test Results Report
	19.20	Certificate of Analysis Print
Master Data Reports	36.17.6	Control Tables Report

Various reports and inquiries associated with signature-eligible menu programs can display signature data. The field that controls this feature—Display E-Signature Details—displays on the user interface based on setup data. See “Functional Reports and Inquiries” on page 108.

The electronic signature function prompts for and maintains signature information based on signature profiles. Each profile is associated with a specific category of data and indicates whether signatures should be captured and for which menu programs, as well as which fields are being signed.

Important Categories are defined by QAD and delivered with the electronic signature functionality. Adding new categories requires custom development.

Electronic Signatures Workflow

Use the programs on the E-Signature Setup Menu to set up and configure electronic signature functions. Figure 6.1 illustrates the electronic signature process workflow; use it to set up signature functions in your environment.

Fig. 6.1
Electronic Signatures Setup Flow

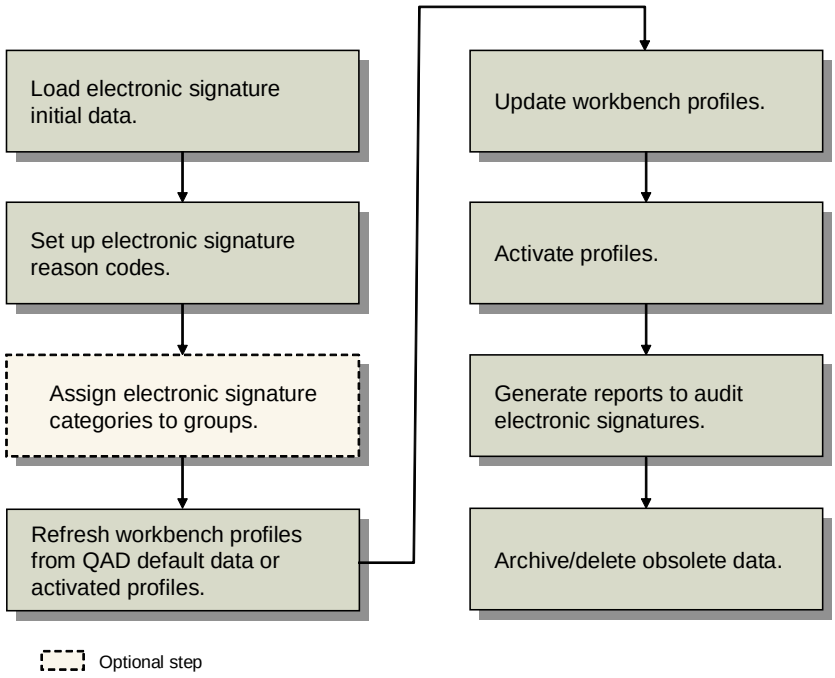


Table 6.2 shows the programs available for setting up and reporting on electronic signature functions.

Table 6.2
Electronic Signatures Programs

Menu Number	Description	Program Name
36.12.4	E-Signature Events Report	esevtrp.p
36.12.5	E-Signature History Report	eshstrp.p
36.12.7	E-Signature Failure Report	esflrp.p
36.12.14.1	E-Signature Group Maintenance	escgmt.p
36.12.14.2	E-Signature Group Report	esgrrp.p
36.12.14.4	E-Signature Workbench Refresh	eswpref.p
36.12.14.5	E-Sig Workbench Profile Maint	eswpmt.p
36.12.14.6	E-Sig Workbench Profile Report	eswprp.p
36.12.14.8	E-Signature Profile Activation	eswpact.p
36.12.14.9	Activated E-Sig Profile Report	esacrp.p
36.12.14.11	E-Sig Category Master Report	escatrp.p
36.12.14.13	E-Sig Initial Data Load	esinild.p
36.12.14.21	E-Sig Failure Archive/Delete	esesigup.p
36.12.14.22	E-Signature Archive/Delete	esesup.p
36.12.14.23	E-Signature Restore	esesld.p

Before electronic signature processing can begin, the prerequisite planning and implementation steps must be completed:

- Planning steps include:
  - Determine the types of data that need to be signed based on the regulatory requirements for your specific industry or environment.
  - Determine how QAD Enterprise Applications fit into your overall business processes, as well as which specific electronic signatures support those processes.
  - Complete data mapping requirements for records and available signatures.
  - Determine security requirements for signed records; for example, assign appropriate role-based security to prevent users who should not sign records from accessing the programs that require signatures.

  Note Electronic signatures should be part of a detailed security plan to meet your overall business requirements.
- Implementation steps include:
  - Define reason codes to explain the meaning of each signature.
  - Optionally, define electronic signature groups to simplify the setup process.
  - Load QAD-provided default signature profiles and modify them as needed, setting appropriate filter criteria.
  - Activate the updated profiles.

The first activity in setting up electronic signature functions is to plan the extent to which you need to require signatures. Regulatory agencies are often specific about the types of data that must be signed, as well as the role of the signing individual—verifier, approver, and so on. Before you start the implementation, be sure that your signatures meet the needs of the appropriate regulatory agency. While the system offers a range of programs, tables, and fields that can be included in signature processing, you might not be required to implement more than a few.

A critical component of virtually any electronic signature is the signature meaning—whether the person applying the signature was approving, inspecting, reviewing, or so on. Reason codes provide the signature meaning. Be sure to plan and implement reason codes that make sense in your specific regulatory environment. See page 92.

To avoid repetitive data entry for individual category profiles, create signature groups in E-Signature Group Maintenance (36.12.14.1). An electronic signature group is a group of category profiles that can be managed at the same time. A category is the definition of a set of system data that can be signed as a unit. Creating an electronic signature group removes the requirement that each category profile must be refreshed or activated individually. When a group is refreshed or activated, profiles for all member categories are automatically updated. This saves time and can be used to organize categories into functionally similar groups. See page 94.

To begin requiring electronic signatures, activate the profiles with E-Signature Profile Activation (36.12.14.8). Activated profiles are staged to begin on a future date; signature recording does not occur immediately after a profile is activated. On the specified begin date, the system begins requiring and recording signature data as defined by each profile. See page 100.

Use E-Signature Events Report (36.12.4) and E-Signature History Report (36.12.5) to view information that applies to electronic signatures. Use E-Signature Failure Report (36.12.7) as part of your security program to identify potential unauthorized access attempts. See page 105.

Categories

A category is a QAD-provided definition of a set of system data that can be signed as a unit in certain menu programs. For example, it identifies a set of tables and fields, as well as the menu program or programs from which this data can be signed.

Because records in a given database table can be updated by more than one program, a category can be associated with more than one menu program. Conversely, a program can update more than one table; multiple categories can apply to a single menu program.

Example The Operation History category (0003) generates signatures for tables and fields that store operation history information. Since these tables can be updated from several Shop Floor Control (menu 16.20) programs, several programs are included in the category. Because those same programs can also update records associated with quality results, they are included in the Quality Results category (0002) as well.

Users cannot update category definitions. Instead, QAD provides a default profile for each category. You can refresh the workbench profiles with these defaults and modify them based on the specific needs of your environment.

Category definitions include a default set of filters that can be used to determine whether a signature is required based on a given value for a site, item number, or other data element. Although filters are defined for each category, their use is optional; control how filters apply to your implementation by updating the category profile using the workbench. See “Filters” on page 90.

Table 6.3 lists the electronic signature categories, as well as the default menu programs associated with them. If you do not want a particular program to generate electronic signatures, you can deselect it in the workbench profile. See “Apply Profile to Menu Programs” on page 98.

Table 6.3
QAD-Defined Categories

Code	Name	Description	Available Menu Programs
0001	InvCtrl	Inventory Control	Inventory Control (3.24) Control Tables Report (36.17.6)
0002	QualRes	Quality Results	Labor Feedback by Work Order (16.20.1) Operations by Work Order Report (16.20.13.14) Operations by Employee Report (16.20.13.15) Operation Transaction Detail Inq (16.20.13.9) Labor Feedback by Employee (16.20.2) Labor Feedback by Work Center (16.20.3) Operation Move Transaction (16.20.6) Test Results Maintenance (19.13) Test Results Report (19.15)
0003	OpHist	Operation History	Labor Feedback by Work Order (16.20.1) Operations by Work Order Report (16.20.13.14) Operations by Employee Report (16.20.13.15) Operation Transaction Detail Inq (16.20.13.9) Labor Feedback by Employee (16.20.2) Labor Feedback by Work Center (16.20.3) Non-Productive Labor Feedback (16.20.4) Operation Complete Transaction (16.20.5) Operation Move Transaction (16.20.6)

Code	Name	Description	Available Menu Programs
0004	ComCtrl	Regulatory Attributes Control	Regulatory Attributes Control (1.22.24)
			Control Tables Report (36.17.6)
0005	LotMstr	Lot Master	Lot Master Maintenance (1.22.1)
			Lot Master Inquiry (1.22.2)
0006	InvDet	Inventory Details	Inventory Detail Maintenance (3.1.1) Detail Maintenance by Item/Lot (3.1.2) Inventory Detail Report (3.6.5) Inventory Detail Maintenance (4.8.16)
0007	InvTran	Transaction History	Inventory Detail Maintenance (3.1.1) Detail Maintenance by Item/Lot (3.1.2) Transactions Detail Inquiry (3.21.1) Transfer–Single Item (3.4.1) Transfer with Lot/Serial Change (3.4.3) Batchload Transfer with Lot/Serial Change (3.4.4) Quality Order Results Entry (19.11)
0008	QualOrd	Quality Order	Quality Order Results Entry (19.11) Quality Order Results Report (19.12) Certificate of Analysis Print (19.20)
0009	PCOInc	PCO Incorporation	Incorporation Selection (1.9.7.4) Incorporation (1.9.7.5) Implementation (1.9.7.13) PCR/PCO Detail Inquiry (1.9.2.8) Print PCR/PCO (1.9.9.1)
0010	PCOAppr	PCO Approval	PCR/PCO Detail Inquiry (1.9.2.8) PCR/PCO Approval (1.9.6.1) Detail Approval Maintenance (1.9.6.13) Print PCR/PCO (1.9.9.1)

Note Some categories are also associated with reports and inquiries that can include electronic signature data. See “Functional Reports and Inquiries” on page 108 for information.

Use E-Sig Category Master Report (36.12.14.11) to view information about the QAD-defined categories.

Category 0007 Considerations

Current signature data for category 0007, Transaction History, is never shown as part of the latest electronic signature when you access a previously signed record from one of the programs listed in Table 6.3 for category 0007. When setting up this category, you should ensure that the fields and filters selected match for programs associated with two categories—such as Inventory Detail Maintenance—to avoid confusion regarding which data the signature is applied to. See “Recording Electronic Signatures” on page 101.

Note You can still view the final data being signed in the final signature data frame for this category.

Category 0006 ld_det Records

Some companies choose to implement temporary locations by setting Permanent to No in Location Maintenance (1.1.18). This setting has consequences for how audit records are created and how electronic signatures occur. This section outlines the effects of auditing and signing temporary location detail records (ld_det).

Inventory Detail Maintenance (3.1.1) and Detail Maintenance by Item/Lot (3.1.2) create the following transactions for non-permanent locations:

- ISS-CHL transaction is created in tr_hist that sets the QOH to zero and deletes the ld_det record.
- RCT-CHL transaction is created in tr_hist that receives the quantity on hand (QOH) for the new inventory detail record that is created for the temporary location.

This standard behavior may lead to some confusion in understanding the audit history because three audit records are created:

- One to delete the temporary ld_det record
- One to create it with all new values
- One to update the QOH

The delete event is for a different ld_det record than the create and modify.

When electronic signatures are enabled, only one signature for these three events is captured. The signature is associated with the last event. This may appear to be misleading in the E-Signature History Report (36.12.5).

If you typically use temporary locations, you should consider this before enabling electronic signatures on this type of record.

Profiles

The electronic signature system maintains signature information based on a signature profile that is associated with a specific category of data. Profiles are identified by the corresponding QAD-defined category codes. The category profile specifies:

- Whether electronic signatures are required
- In which programs
- Which fields are signed
- Characteristics of how signatures are displayed and recorded
- Filter definitions

The life cycle of a profile consists of three phases:

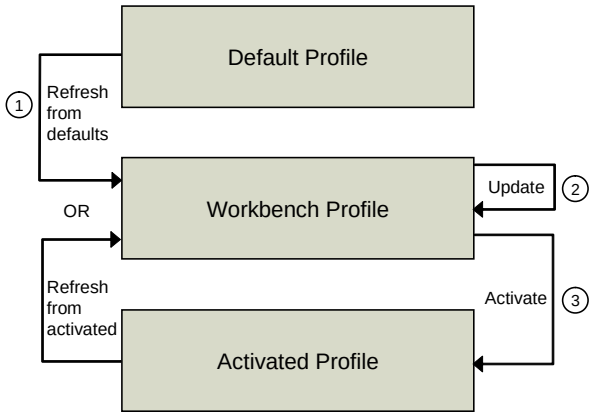
- The QAD-provided default profile. Based on QAD-provided category data, this is loaded using E-Sig Initial Data Load (36.12.14.13) and serves as the template for profiles used by the system. You cannot update default profile records directly—only after you have copied them by refreshing the workbench profiles. See “Refreshing Signature Profiles” on page 94.

  Note You can view the structure of default profiles without refreshing the workbench. Use E-Sig Workbench Profile Report (36.12.14.6) with Display Default Profile set to Yes.

- The workbench profile. This is initially based on the corresponding default profile for a given category. It is an intermediate working version used to tailor each profile for specific requirements. You can refresh it based on an existing activated profile or the default profile. Because the workbench profile has no effect on current system activities, you can continue to update it while the active version controls electronic signature processing. See “Updating Signature Profiles” on page 95.
- The activated profile. This is the profile used by the system to control electronic signature processing. It is copied from the workbench profile during activation along with a begin date, and it stays in effect until the begin date of another active profile for the same category. See “Activating Electronic Signature Profiles” on page 100.

Figure 6.2 summarizes the relationships between the three category profile types.

Fig. 6.2
Profile Flow



Tables and Fields

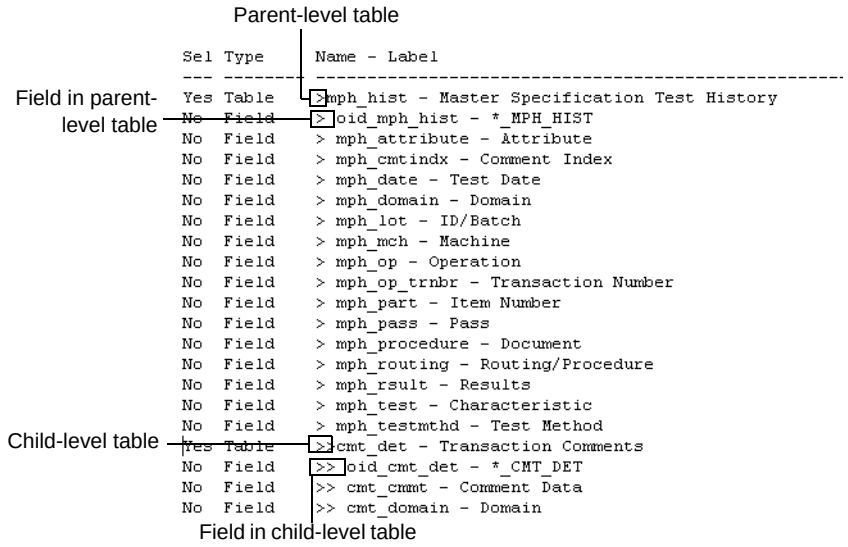
The category profile includes a list of tables and fields that define the data to be signed in the corresponding signature-enabled programs. See “Updating Signature Profiles” on page 95.

Each category profile includes one or more database tables and their corresponding set of fields. For example, the profile for category 0007, Transaction History, includes fields from the inventory transaction history table (tr_hist). In some cases, a category profile might include multiple tables where the records are related in a hierarchy of parent-child relationships. For example, a table might have associated child records in the transaction comments (cmt_det) table.

Greater-than symbols (>) and spaces show the hierarchical relationships among tables and fields on the list. Top-level tables are preceded by a single > symbol; fields within the table begin with a > symbol and a space. Tables with child relationships are designated with an additional > symbol; fields in child tables include the same number of > symbols as the corresponding tables, again with a space separator.

Example Figure 6.3 shows a portion of the default profile structure for category 0002, Quality Results, which specifies the test results data to be signed in several programs in Shop Floor Control (menu 16.20). View default profiles using E-Sig Workbench Profile Report (36.12.14.6) with Display Default Profiles set to Yes.

Fig. 6.3
Example of Workbench Profile Table/Field Structure



Top Tables

Each QAD-provided category definition includes a top-level table, which displays in the Top Table field in the first frame of E-Sig Workbench Profile Maintenance. In most cases, this is the first table that appears in the profile structure.

In other cases, however, the top table is not included in the data to be signed but instead provides key values for identifying the signed data.

Example The top table in the Quality Results category is the work order routing (wr_route) table, but this table is not included in the data to be signed; that consists of the master specification history (mph_hist) table and related transaction comments (cmt_det). The wr_route record is used only to identify the signed data by providing the context.

You can specify top-table field values to identify data that may have signatures attached; for example, use E-Signature History Report (36.12.5) to view signature history associated with a specific work order identified in the wr_route table. See “Electronic Signature Reports” on page 105.

Filters

Depending on the specific requirements of your environment, you may not need to record electronic signatures for all records of a given type. For example, you might want to require signatures only on inventory transactions involving a specific site or certain items.

QAD-provided categories include filters for selecting or excluding data that must have electronic signatures applied.

Table 6.4 indicates the filters that are available in each QAD-provided category definition.

Table 6.4
Available Filters, by Category

	Filter				
Category	Domain	Site	Item Number	Location	Work Center
0001 Inventory Control	✓				
0002 Quality Results	✓	✓	✓		✓
0003 Operation History	✓	✓	✓		✓
0004 Regulatory Attributes Control	✓				
0005 Lot Master	✓		✓		
0006 Inventory Detail	✓	✓	✓	✓	
0007 Transaction History	✓	✓	✓	✓	
0008 Quality Order	✓	✓	✓	✓	
0009 PCO Implementation	✓				
0010 PCO Approval	✓				

When you refresh a workbench profile based on the QAD-provided default profile, the filter mode is set to indicate that filtering will not be applied. If you choose to set up signature requirements based on available filters, specify appropriate values when you define your implementation-specific profile in E-Signature Workbench Profile Maintenance. See “Set Up Filters” on page 99.

Filters are designed to work either by inclusion or exclusion, as defined by the Filter Mode field in E-Signature Workbench Profile Maintenance. For example, an *inclusion* filter might be set up to include records by site and location. If you set up the filter criteria with site values of 1000 and 2000 and location values of loc1 and loc2, only records with a combination of one of those sites and one of those locations will require an electronic signature. In this scenario, updating a record associated with site 1000, loc3 would not trigger a prompt for an electronic signature.

In the same example, defined as an *exclusion* filter, electronic signatures would not be required for records with any combination of the specified sites and locations. Updates to records with any other sites and locations, however, would trigger a signature prompt.

A profile can have either inclusion or exclusion filters, but not both.

Completing Prerequisite Activities

Before you start setting up records that control when electronic signatures are required and how they are recorded, you should complete the following tasks:

- Load electronic signature initial data
- Define signature reason codes
- Check Security Control settings

Load Electronic Signature Initial Data

The initial data, QAD-provided default profiles, must be loaded into the system first. Use E-Sig Initial Data Load (36.12.14.13) to load the data.

Enter the directory of the files and all the default profiles in that directory will be loaded into the system. These files are located on the server side under *installdir*/mfg, where *installdir* is the root application installation directory.

Defining Signature Reason Codes

The signature reason code is a critical element of the electronic signature. In regulatory environments, the signature record typically must include the meaning of the signature. The system uses reason codes to provide the meaning.

Each time the system prompts for an electronic signature, the user must provide a valid reason code. For example, reason codes might indicate that a quality record has been approved, reviewed, or inspected. See “Recording Electronic Signatures” on page 101.

Use Reason Codes Maintenance (36.2.17) to define signature reason codes that are appropriate to your environment.

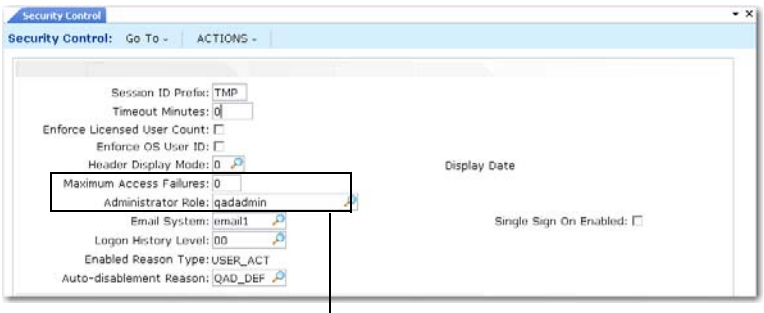
Important All reason codes used by electronic signatures must be associated with the QAD-provided ESIG reason type. Reasons of any other type cannot be entered in the signature prompt frame.

Reviewing Security Control Settings

To guard against attempts by unauthorized individuals to apply electronic signatures using another user’s ID, signature validation logic is similar to that used in the login process. See Chapter 2, “Security Overview,” on page 11 for information on setting up and using login security.

Review Security Control (36.3.24) to see how login security is defined in terms of password structure and use rules.

Fig. 6.4
Security Control (36.3.24)



Two fields directly control how the system manages unsuccessful electronic signature attempts:

- Maximum Access Failures indicates how many consecutive unsuccessful signature attempts cause the user’s session to terminate, disable the account, and inform the administrator role of a potential unauthorized access attempt.

- Administrator Role is the name of the role—defined in Role Create (36.3.6.1)—assigned to the system users who are notified by e-mail when a session is terminated because of excessive unsuccessful signature attempts. The system also sends e-mail to users who are assigned this role when a signature profile is activated. See “E-Mail Notifications” on page 103.

Defining Electronic Signature Profiles

Setting up and using electronic signature profiles include these steps:

- Create electronic signature groups.
- Refresh workbench profiles.
- Update workbench profiles.
- Activate profiles.

Overview

Each category is associated with one or more signature-eligible programs in its own profile. Initially, all signature profiles are empty; they must be refreshed with the QAD-provided information. Category profiles hold values that electronic signature functions use to manage the information retention and reporting process. This information affects electronic signature functions only after the profile is activated.

A category profile:

- Indicates whether signature functions are enabled for the category in general and for specific menu programs.
- Specifies control information that determines how electronic signature data displays when an enabled program runs.
- Maintains a list of tables and fields that defines the data to be signed. This data is included in signature records.
- Defines filters that can be used to determine whether electronic signature requirements apply to all records or only those containing specified values.

The system maintains three sets of profiles: the QAD-supplied default profiles, the profiles you edit in the workbench, and the activated profiles. See “Profiles” on page 88. When you activate a profile, the system creates a new activated profile by copying your completed workbench profile and setting the begin date. Since the system activates a copy of your workbench profile, you can continue to modify the workbench profile with E-Signature Workbench Profile Maintenance without affecting the active system.

Before refreshing workbench profiles, you can optionally create signature groups to manage several profiles more easily and streamline the data setup process. Once refreshed, modify the workbench profiles with your requirements. You can enable or disable signatures and update filters as needed. When your workbench profiles are complete, activate them and set a begin date. To discontinue signatures, simply update the workbench profile to set E-Signature On to No; then activate it with the begin date set to the date signatures are no longer needed.

Creating Signature Groups

Use E-Signature Group Maintenance (36.12.14.1) to group all the categories you plan to control using electronic signatures, or to group related categories for signature purposes. Signature groups streamline the setup process by letting you refresh and activate the profiles for all member categories at once, instead of one profile at a time.

Example You might create a group called Control that includes the Inventory Control (0001) and Compliance Control (0004) categories so that you can refresh and activate both control program-related profiles at the same time.

Specify a group name, up to eight characters. An electronic signature group cannot have the same name as a category code.

Next, provide a brief description and choose Next to display the Group Detail frame, which lists all the categories currently assigned to the group. Use the Cross Reference Maintenance frame to add or delete categories.

Use E-Signature Group Report (36.12.14.2) to display the records defined in this program.

Refreshing Signature Profiles

When initially setting up electronic signature functions, workbench category profiles are empty and must be manually populated. Use E-Signature Workbench Refresh (36.12.14.4) to update the empty profiles with the QAD-provided default information. You can refresh one category at a time or, optionally, refresh the profiles for an entire group of categories.

You can use this program later to restore the QAD-provided default data, modified in E-Signature Workbench Profile Maintenance, or to update workbench profiles based on existing active profiles.

Note Any changes you make with this program do not affect activated profiles currently in use.

Indicate if you want to refresh categories or groups; then use the Value field to specify the category name or group name to be refreshed. Leave Value blank to refresh all categories or groups, based on the setting in the Group/Category field. If Value is blank, the system prompts you to confirm.

Use the following field descriptions to enter the values for the refresh process.

Refresh Profiles. Indicate whether to refresh all data for the specified profiles. When this field is Yes, an additional frame displays that you can use to determine which profiles are used as the source of the updates.

Override Fields. Indicate whether to override the field that controls electronic signatures for the specified profiles. When this field is Yes, an additional frame displays.

Refresh Profile Frame

If Refresh Profiles is Yes, the Refresh Profile frame displays.

Source Profile. Enter Activated or Default to indicate which profiles to use as the source for refreshing the profiles selected previously.

Activated: Each specified workbench profile is refreshed using the activated profiles in use on the date specified in Effective Date. The corresponding profiles must be in use on the date specified; otherwise, the system displays an error for each activated profile not found and the refresh does not occur for that profile.

Default: Each specified workbench profile is refreshed using the QAD-provided values. Select this value when initially setting up electronic signature functions to load the QAD-provided values into the profiles for the categories in which you plan to use signatures.

Effective Date. Enter a date when the activated source profile was in use. The workbench profile is refreshed using the active source profile settings in use on this date. If an activated profile was not in use on the specified date, an error displays and the target profile is not refreshed.

Note This field is available only when Source Profile is Activated.

Example Enter today's date to refresh the workbench profiles based on the activated profiles currently being used.

Override Fields Frame

If Override Fields is Yes, the Override Fields frame displays.

E-Signature On. Indicate whether to enable electronic signature functions for the profiles being refreshed.

If Refresh Profiles is No, the value specified here replaces the E-Signature On value in the current workbench profiles for the specified group or category. However, no other workbench data is updated.

When you refresh based on QAD-provided profiles, signature functions are turned on by default. You can use this field to override that setting.

Use E-Signature Workbench Profile Maintenance to change this value for individual profiles.

Updating Signature Profiles

Use E-Signature Workbench Profile Maintenance (36.12.14.5) to adjust profile settings for your specific environment by:

- Defining control settings that determine how electronic signature processing works for each category
- Specifying the menu programs from the available list where signatures will be applied to the category
- Updating the list of tables and fields that are to be signed and included in signature records
- Setting up filters to control whether specific data is subject to or exempt from signature requirements

To disable electronic signatures for a profile that currently requires them, you must create a new activated profile for the category. Do this by updating the workbench profile and setting the E-Signature On value to No; then activate that new profile with the proper begin date. See "Activating Electronic Signature Profiles" on page 100.

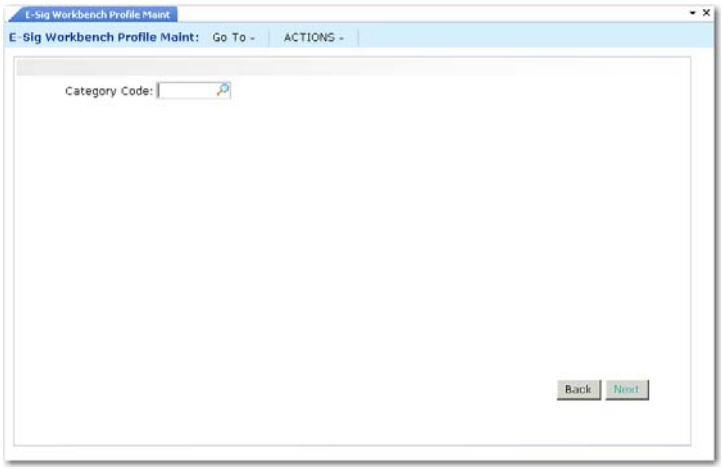
Use E-Signature Workbench Profile Report (36.12.14.6) to display the information updated in this program.

Note Some special considerations apply when you are setting up profiles that involve category 0007, Transaction History. See page 87 for information.

Specifying Control Settings

Figure 6.5 illustrates the first frame of E-Signature Workbench Profile Maintenance.

Fig. 6.5
E-Signature Workbench Profile Maintenance (36.12.14.5)



Enter a QAD-defined category code and choose Next. The system displays several fields you can use to control electronic signature processing.

Top Table Name. The system displays the name of the table used to identify the set of data defined by the category; this sets the context for the signed data.

Example Category 0002, Quality Results, has a value of wr_route (work order routing) in this field. Master specification test history (mph_hist) is shown as the first table in the 0002 profile structure. One electronic signature could contain many records of this type—so the mph_hist identification is not unique. However, all mph_hist records from the electronic signature instance are related to a single wr_route record, which serves as a unique identifier for the signed data. See “Tables and Fields” on page 89.

E-Signature On. Indicate whether the system should apply the electronic signature functions for the category defined in this profile when it is activated.

No: Electronic signatures do not apply to this category. Use this option to turn electronic signatures off for programs that currently require them. For example, if signatures are currently used and a new profile for this category with E-Signature On set to No is activated, electronic signature functions stop on the new profile’s begin date.

Yes: Once this profile is activated, electronic signatures are required for this category as defined by the menu details and applicable filters.

When you refresh from QAD-provided default data, the value is Yes.

Display Latest E-Sig. Indicate whether the system displays the latest electronic signature when programs controlled by this profile are executed. See “Recording Electronic Signatures” on page 101.

When you refresh from QAD-provided default data, the value is Yes.

Prompt for Preview E-Sig. For programs that generate transactions, enter Yes to have the system prompt for a signature before the transaction data is created. The user is given the option to display the final data before signing. You can use this feature to avoid potential record-locking issues. This feature does not apply to all signature-enabled programs.

When the user sets Show Final Data to Yes when entering a signature, the system creates the transactions and displays final data before it is signed. Otherwise, the user enters the signature without viewing the final data.

When you refresh from QAD-provided default data, the value depends on the types of programs included in the category.

This configurability is provided to address record-locking issues that might be caused by the user interacting with the signature frame. In some menu programs that create transaction records such as operation or transaction history, the system locks frequently updated records while creating the transaction records. These programs have been designed to minimize the amount of time that records are locked by having no user interaction during record creation.

When electronic signatures are used with these programs and the final data to be signed—including the transaction data—must be displayed to the user while prompting for the signature, records remain locked until the user successfully completes the signature. This record-locking during signing is necessary because all changes must be rolled back if the signature is not accepted. During this time, no other users can update these same locked records. This issue becomes even more problematic, for example, if the user decides to leave their computer at this crucial time, before entering the signature fields.

This problem can be avoided in most situations because the relevant data for the user to review before signing are the fields that the user entered. These fields are generally available in the preview signature frames. After the signature is accepted, the program generates the transaction records and includes them in the signed data stored with the signature. Your system validation process can provide the assurance that the program systematically and reproducibly generates the transaction records based on the entered data. So, by signing in the preview signature frame, the final data never needs to be displayed and the records will not be locked any longer than required to create them. If the signature is not accepted, all user changes are rolled back and the transaction records are not created.

Set Prompt for Preview E-Sig to Yes to avoid these potential problems.

Data Frame Optional. Enter Yes to allow users to immediately enter an electronic signature without scrolling through the data to be signed. In this case, they can still view all the fields by setting Scroll Details to Yes in the signature frame.

When the field is No, focus is on the frame that displays the data to be signed. To enter the signature, users must first choose End to exit that frame.

When you refresh from QAD-provided default data, the value is Yes.

Prompt for Remarks. Indicate whether the user can add an optional remark while entering electronic signature data. When this field is Yes, a 64-character updateable Remarks field displays in the signature frame. Remarks are included in the electronic signature record.

When you refresh from QAD-provided default data, the value is Yes.

Filter Mode. Specify the type of filtering the system will use in determining whether specific data requires electronic signatures. See “Filters” on page 90.

None: Filters are not used. The Filters and Filter Criteria frames do not display.

Inclusion: Only data meeting the specified filter criteria requires electronic signatures.

Exclusion: All data except those meeting the specified filter criteria require electronic signatures.

Note A profile can have either inclusion or exclusion filters, but not both.

When you refresh from QAD-provided default data, the value is None.

Multiple Categories

Based on the data they update, some menu programs can be associated with more than one category. When this occurs, the system includes logic to resolve conflicting workbench profile setup data for three settings:

- Prompt for Preview E-Sig
- Data Frame Optional
- Prompt for Remarks

Table 6.5 shows the sequence the system uses for determining which profile takes precedence in each such case.

Note This logic is needed only when a program is selected in the Workbench Profile Menu Details frame of more than one category profile. Additionally, when the menu program is executing, if a signature is not required for the first category, the second category profile is used to determine these three settings.

Table 6.5
Profile Precedence for Multiple Categories

Menu Program	Category Sequence
Labor Feedback by Work Order (16.20.1) Labor Feedback by Employee (16.20.2) Labor Feedback by Work Center (16.20.3) Operation Move Transaction (16.20.6)	1. Operation History (0003) 2. Quality Results (0002)
Quality Order Results Entry (19.11)	1. Transaction History (0007) 2. Quality Order (0008)

Apply Profile to Menu Programs

When you initially set up electronic signature functions by refreshing profiles based on QAD-provided data, each category is associated with one or more menu programs that update the data defined in the category.

Although you cannot specify additional programs, you can use the Workbench Profile Menu Details frame to control whether signature functionality will apply to the available menu programs.

When a program is included in the category profile, an asterisk (*) displays in the Apply column. Clear the field to deselect a program.

Note If a program appears more than once in the menu system, the frame lists all menu numbers. Changing the Apply setting for one menu number automatically updates all.

In some profiles, the program list includes reports and inquiries. See “Functional Reports and Inquiries” on page 108. These programs can display signature data if included in the activated profile. When they are included, they have a Display E-Signature Details field that gives the user the option of displaying signature data in the output.

Select Tables and Fields

QAD-provided setup data includes a set of tables and fields that define the data to be signed and stored with the signature. The Workbench Profile Structure frame lists the tables and fields defined by the category.

If the current profile was refreshed based on default data, all tables and fields are selected.

Toggle the asterisk in the Sel column to select or deselect fields or tables. If you deselect or select a table, all fields in the table are automatically deselected or selected as well. In that case, the frame display does not refresh immediately.

Note The first field listed for each table is the system-assigned object ID (OID) that uniquely identifies each record in the database. You cannot deselect this field.

The system uses greater-than symbols (>) and spaces to show the hierarchical relationships between table and field elements in the profile structure. See “Tables and Fields” on page 89.

Set Up Filters

When Filter Mode is Inclusion or Exclusion in the Workbench Profile Details frame, additional frames let you select and set up filters. Filter frames do not display when Filter Mode is None.

These settings determine whether electronic signature processing occurs for data associated with specified values. See “Filters” on page 90.

Use the Filters frame to specify which of the available filters you want to apply to this category profile. When the Sel column includes an asterisk, the filter is selected and displays in the Filter Criteria frame.

Note You cannot complete the profile record if all selected filters do not have at least one criteria value. The system prompts you to remove such filters from the profile.

The Filter Criteria frame lists all the filters that were selected in the Filters frame. To enter criteria values for a filter, navigate to the Criteria Value frame and enter a value that will be used to either include or exclude electronic signature processing, depending on the filter mode.

You cannot enter data ranges for a filter. Instead, enter multiple criteria values. Each criteria value displays on a separate line in the Filter Criteria frame.

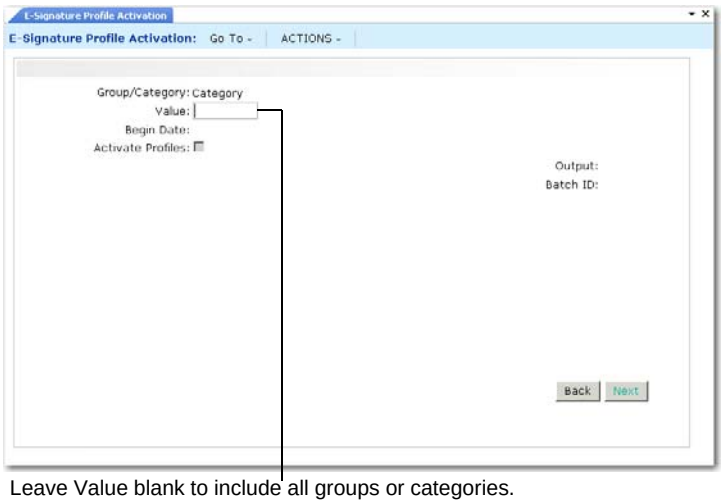
To filter on a blank value, enter the filter field name and leave Value blank. The system prompts you to confirm. A blank value is not a wildcard; instead, it only matches data where the value is actually blank.

Important Since the system does not validate this value, you should be careful when you set up filters. For example, if you are setting up an inclusion filter to require electronic signatures only for a single site and accidentally enter an invalid site code, the program will never prompt for a signature.

Activating Electronic Signature Profiles

After completing the workbench profiles, use E-Signature Profile Activation (36.12.14.8) to activate profiles for one category or a group of categories. Activated profiles are staged for electronic signature functions to begin on a future date; signature settings are not in effect immediately after a profile is activated.

Fig. 6.6
E-Signature Profile Activation (36.12.14.8)



Profiles cannot be activated on the begin date. Plan all changes ahead of time and activate updated profiles before their begin date. Profiles must have the begin date set to sometime in the future. Activated profiles become effective at 12:00 AM on the specified date.

You can execute this program in batch mode if you are activating a group with many associated categories.

When this program completes execution, it generates a report that displays information for each activated profile. The report includes the following for both the original profile and the newly activated one:

- The category name.
- The value of E-Signature On.
- The begin date.
- The data structure of the profile, listing all tables and fields that are marked as selected in E-Signature Workbench Maintenance. The system uses greater-than symbols (>) and spaces to show the hierarchical relationships between data elements. See “Tables and Fields” on page 89.

If Activate Profiles is No, only the report is generated; the profiles currently in use are not updated. You can use this setting to verify the effects of running the program before you actually activate the profiles.

Use Activated E-Sig Profile Report (36.12.14.9) to display details about activated profiles.

When a profile is activated, the system automatically sends an e-mail message to system users who are assigned to the administrator role in Security Control (36.3.24). See “E-Mail Notifications” on page 103.

Recording Electronic Signatures

When profiles with E-Signature On set to Yes have been activated using E-Signature Profile Activation and the specified begin date is reached, the system automatically begins prompting for electronic signatures based on rules defined in the active profile.

When Display Latest E-Sig is Yes in the active profile, before displaying data defined by the category, the system displays the signature that was recorded most recently for that data.

Note Latest signature data for category 0007, Transaction History, is not included in the display for programs associated with that category. See page 87.

The top frame of a signature display includes such information as the user ID and name of the person who applied the signature and the associated reason code. Event ID is a system-assigned identifier for a specific electronic signature.

The signature display also includes a Current field, which indicates if all the signed data fields recorded at the time of the signature still have the same values. If an included field has been updated since the record was signed—for example, with another program that is not signature enabled—the system sets Current to No.

Note The Current setting is not stored as part of the signature instance. It is determined in real time based on the activated profile currently in effect. If multiple categories are signed in one menu program, each category of signed data is independent of the others. If the data changes in one, it does not affect the Current setting of the others.

The lower frame shows the value of the signed data fields at the time of the last signature. Greater-than symbols (>) and spaces show the hierarchy of the data structure. See “Tables and Fields” on page 89.

Note If the data about to be displayed has never been signed, the system displays a message for the associated category.

You can scroll through the frame to view all the field values. Choose End to exit from the details frame and return to the program.

When you finish entering or updating data according to the standard menu program functionality, the system prompts you to enter an electronic signature.

Note The points at which a program saves updates to the database may change when electronic signatures are enabled. See “Transaction Scoping” on page 102.

The prompt screen includes the signature frame, as well as a details frame showing the data being signed.

Navigation in the details frame depends on the setting of Data Frame Optional in the active profile. When that field is No, focus is immediately on the details frame so you can scroll through the entire record. You must choose End to place focus on the signature frame. When Data Frame Optional is Yes, immediate focus is on the signature frame. However, you can still scroll the details by setting Scroll Details to Yes. When you finish reviewing the list of field values, choose End to return to the signature frame.

In menu programs that create transaction records, these signature frames may display before the transaction records are created, depending on the value of Prompt for Preview E-Sig in the activated profile. See “Prompt for Preview E-Sig” on page 97. In this case, the user can choose to complete the signature based on the incomplete data displayed in the details frame by setting Show Final Data to No. The transaction records are created, and the signature is recorded along with values for all signed fields, including the transaction record fields.

To see the final data to be signed including the transaction records, set Show Final Data to Yes. The system generates the transaction records and displays the signature and details frames.

To sign the data, you must enter your user ID, password, and a valid reason code defined for reason type ESIG. Note that the User ID field must be the same as your system login ID. Depending on the Prompt for Remarks field in the active profile, you may also be able to enter a remark related to the signature.

If you choose not to sign or the signature is not accepted, the system rolls back the entire database transaction, including all user modifications.

Important Be careful to enter the same user ID you used for login, as well as the correct case-sensitive password. Based on settings in Security Control (36.3.24), too many invalid signature attempts can cause your session to terminate, disable your user ID, and inform the system administrator of a potential unauthorized access attempt. See “Reviewing Security Control Settings” on page 92.

Depending on how security is set up in your system, the system may prompt you to change your password. For example, this can happen if the password has reached its expiration date while you were logged in, or if the system administrator has forced a password change for your user ID.

After signature processing is completed, the system displays a message indicating that the signature has been successfully executed, along with the event identifier.

Transaction Scoping

So that the system can apply electronic signatures to the appropriate data, transaction scoping—the points during program execution when data is committed to the database—has been modified in some maintenance and transaction programs that can be signature enabled. See “Apply Profile to Menu Programs” on page 98.

For example, before electronic signature functionality was added, each frame in Inventory Control (3.24) was included in an individual transaction block. You could update the first frame, choose Next, then choose End from the second frame. The system updated the database with the changes to the first frame. You did not have to choose Next through all the frames.

However, all frames are now part of one transaction block—allowing the system to apply the same electronic signature to all updates made in the program. If you update the first frame, choose Next, and choose End in the second frame, the changes you made in the first frame are not saved to the database. You must choose Next through all the frames to save any changes you make in the program.

Product Change Control

If you use electronic signatures with the Product Change Control (PCC) menu, Incorporation (1.9.7.5) and Implementation (1.9.7.13) do not behave the same way as other signature-enabled programs.

Because all product change orders (PCOs) that are available for incorporation or implementation are selected by the system and processed only once, no current signature record is ever available for display when one of these programs executes. Additionally, the programs do not display the records being signed. Instead, the system just prompts for an electronic signature for each PCO to be incorporated or implemented.

Each PCO is processed in one transaction. If an error occurs during incorporation or implementation processing, all data related to this PCO is rolled back—including updates to product structures, routings, and so on. Other PCOs processed in the same program session are not affected.

If the user presses End in the E-Signature frame, the system does not create an electronic signature, and rolls back the incorporation or implementation transaction for the PCO. It then continues to process the next PCO.

Note You cannot use batch processing with Incorporation or Implementation when electronic signatures are enabled for the program. The Batch ID field does not display.

E-Mail Notifications

The system generates and sends e-mails to the system users who are assigned the administrator role in Security Control (36.3.24) in the following situations:

- One or more signature profiles are activated.
- A user's consecutive number of failed electronic signature attempts exceeds the Maximum Access Failures value in Security Control.

For more information see "Reviewing Security Control Settings" on page 92.

The e-mail text is defined in master comment data. You can customize this text for your environment by modifying the text using Master Comment Maintenance (2.1.12).

The electronic signature-specific messages have a comment type of ES. The comment reference varies depending on the specific purpose. The e-mail is constructed by starting with a specific comment, followed by one or more messages with additional details. A generic comment of type AT with a reference of `email_postfix` is appended. This comment contains the following information that applies to all system-generated security and enhanced controls e-mails:

```
This email was automatically generated from a QAD process. If you have any questions
about this E-mail, contact the QAD system administrator. Do not reply to this E-mail.
```

Signature Profile Activation E-Mail

Comment Reference: email_esig_profile_activation

Comment Type: ES

The e-mail sent for signature profile activation is similar to this example.

```
The purpose of this e-mail is to inform you that one or more e-signature categories has
been activated. You have been included in this e-mail distribution because you belong
to the Administrator role identified in Security Control. The information listed below
regarding the activation can be used to obtain a detailed report of the activation by
running the Activated E-Sig Profile Report.

The activation was performed by User ID: XXX

The newly activated profiles are set to begin on date: dd/mm/yy

The number of newly e-signature enabled activated profiles: #

The number of newly e-signature disabled activated profiles: #

This email was automatically generated from a QAD process. If you have any questions
about this E-mail, contact the QAD system administrator. Do not reply to this E-mail.
```

Signature Failure E-Mail

Comment Reference: email_failed_esig_prefix

Comment Type: ES

The e-mail sent to system users who are assigned the administrator role when failed signature attempts exceed the Security Control value is similar to this example:

```
The purpose of this e-mail is to inform you a user has been disabled for exceeding the
maximum e-signature failures allowed as set up in Security Control. You have been
included in this e-mail distribution because you belong to the Administrator role
identified in Security Control.

User ID deactivated for exceeding max e-sig failures allowed: XXX

This email was automatically generated from a QAD process. If you have any questions
about this email, contact the QAD system administrator. Do not reply to this email.
```

Reporting

Reports and inquiries related to electronic signatures are available in three areas:

- Setup
- Electronic signature reports
- Functional reporting for programs that are signature enabled

Setup Reports

The E-Signature Setup Menu has four reports that provide information on signature setup records:

- Use E-Sig Category Master Report (36.12.14.11) to view the top-table name and the filters available for categories.
- Use E-Signature Group Report (36.12.14.2) to view the categories assigned to each group.
- Use E-Sig Workbench Profile Report (36.12.14.6) to view the following kinds of information about the current workbench structure for a specified electronic signature category:
  - Settings that control processing and display of signatures in enabled programs
  - The list of programs that are signature enabled for the category
  - The list of field and tables that are included in the signature record
  - Optionally, information about filters associated with the category, if applicable

Note Depending on whether you have updated or refreshed a workbench profile since last activating it, this report does not necessarily show the settings currently in use for a category. Use Activated E-Sig Profile Report to view that information.

- Use Activated E-Sig Profile Report (36.12.14.9) to view information about profiles that have been activated using E-Signature Profile Activation. It displays the same types of information as E-Sig Workbench Profile Report, but lets you specify a range of categories over a range of effective dates.

  Example To view all the profiles currently in use, leave the category code range blank and enter today's date in both date fields.

  Note Although a date range is not required in the selection criteria, consider entering one. This significantly reduces the time required to generate the report.

Electronic Signature Reports

The Enhanced Controls Menu includes three reports that let you:

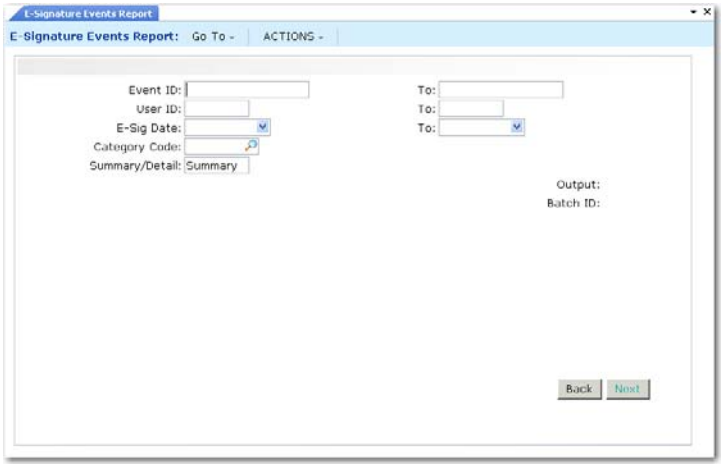
- Display signature events based on information related to the signature itself, such as the user, date, and meaning.
- Select database records based on ranges of values for fields in category top tables, and generate a report on related electronic signature history.
- Monitor login history records for failed electronic signature attempts.

Viewing Signature Events

Use E-Signature Events Report (36.12.4) to view data based on ranges of signature event IDs, user IDs, and dates when the signature was created. Optionally, you can limit the report to signatures related to a single specified category code.

The Summary/Detail field controls whether the report includes just basic information such as the user's name, date, and signature meaning, or also includes details of the signed data.

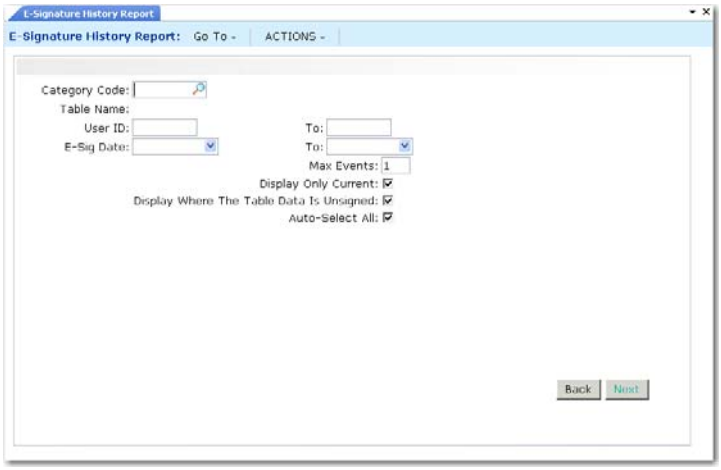
Fig. 6.7
E-Signature Events Report (36.12.4)



Viewing Signature History

Use E-Signature History Report (36.12.5) to select database records and view historical electronic signature data associated with them. For example, you can report on the two latest signature events associated with a specified work order.

Fig. 6.8
E-Signature History Report (36.12.5), Initial Frame



This report includes multiple frames. First, specify the category, user ID range, and signature date range. Category is a required field. Use the following fields to control other characteristics of the report:

Max Events. Specify the maximum number of electronic signature events to be included in the report for each selected record. The default is 1, which displays the latest signature event for each record that matches the data ranges in the E-Record Selection Criteria frame. If you enter a larger number, the system displays the latest first, then works backward through the number of events specified.

Display Only Current. Indicate whether the system should limit the selection to records in which no data has been updated since the latest electronic signature was recorded.

Display Where the Table Data Is Unsigned. Indicate whether the system should include records matching the criteria data ranges even if they are not covered by an electronic signature instance. When this is Yes, the output identifies records that do not have associated signatures.

Auto-Select All. Indicate if you want all the fields in the top table to be included in the report by default. You can modify the setting for individual fields as needed in the Report Display Fields frame. The default is Yes.

Press Next to display the E-Record Selection Criteria frame where you can identify the records for which you are interested in seeing signature histories. Specify ranges of values for one or more fields in the top table for the category.

Note Large reports may result if you do not specify field-level selection criteria.

This frame displays the name, label, and type for each field in the top table of the selected category. Field types are Primary (P), Indexed (I), or non-indexed (F). Any selection criteria entered in the Data Range frame display next to the corresponding field on the E-Record Selection Criteria frame. These selection criteria are used to narrow the search results. See “Top Tables” on page 90.

To minimize the report output, enter criteria for as many table fields as needed. For example, if you are reporting signature records for the Quality Results category (0002), you can limit the report to signatures for a specific work order. Scroll to the Work Order (wr_nbr) field and press Next. Enter the work order number in both the From Value and To Value fields. After entering the field-specific selection criteria for your report, choose End to continue.

Use the Report Display Fields frame to select or deselect the top-table fields to include or exclude on the resulting output.

All fields are preselected if Auto-Select All is Yes in the first frame. Select or deselect fields as needed. Then press Next to specify the output device for the report.

The report output includes the values for all the top-table fields selected in the Report Display Fields frame, as well the following signature data for each event:

- Event ID
- User ID and name of the person signing
- Name of the menu program that generated the signature
- Signature meaning—the reason code entered when the record was signed
- Signature date and time
- Remark entered with the signature
- Current indicator, specifying whether signature values and database values are still identical
- Signed data—the value when signed of each field included in the active profile in effect when the signature was created

Note If signature events are not available that match the selection criteria, the output includes the following message:

```
Data archived or never signed
```

The latest signature should always be available. It is not deleted during an archive/delete.

Monitoring Failed Signature Attempts

As part of an overall security program, you can generate a report showing unsuccessful signature attempts, based on user login history records.

Use E-Signature Failure Report (36.12.7) to select history records by a combination of user ID, signature attempt date, and status code. The resulting report displays the user ID and name, time data, and the status code, which identifies the reason for failure; for example, ID disabled because of excessive failed signature attempts.

When failed login history records are no longer needed online, you can remove them using E-Sig Failure Archive/Delete (36.12.14.21). This standard archive/delete program deletes records from the system and optionally saves them to a file named `esig_fail_YYYYMMDD.hst` where *YYYYMMDD* is the date you run E-Sig Failure Archive/Delete. If this function runs multiple times a day, the data will be appended to the same file for the given day.

Functional Reports and Inquiries

Some reports and inquiries associated with signature-enabled menu programs let you include electronic signature data in the output. When Display E-Signature Details is Yes, the system displays information about the signature such as the individual who signed the record, as well as values of the signed data fields.

Note The display of this field is conditional. It only appears on the user interface when both the following are true in the active profile for the appropriate category:

- E-Signature On is set to Yes. See page 95.
- The menu program has Apply selected. See page 98.

Based on those values, the reports and inquiries listed in Table 6.6 can include the Display E-Signature Details field.

Table 6.6
Reports and Inquiries Displaying Electronic Signature Data

Program	Menu	Category
PCR/PCO Detail Inquiry	1.9.2.8	0010
Print PCR/PCO	1.9.9.1	0009
Lot Master Inquiry	1.22.2	0005
Inventory Detail by Lot Inquiry	3.1.13	0006
Inventory Detail by Item Browse	3.2	0006
Inventory Detail by Site Browse	3.3	0006
Inventory Detail Report	3.6.5	0006
Inventory Detail by Location	3.6.6	0006
Inventory Detail Report	3.6.5	0006
Transactions Detail Inquiry	3.21.1	0007
Operation Transaction Detail Inquiry	16.20.13.9	0003
Operations by Work Order Report	16.20.13.14	0003
Operations By Employee Report	18.4.14	0003
Quality Order Results Report	19.12	0008
Certificate of Analysis Print	19.20	0008
Control Tables Report	36.17.6	0001 or 0004[1]

1. The signature details field displays in Control Table Report if the profile conditions are met for either category.

Important In some inquiries, if Output is set to a display device such as Terminal rather than to a printer or a file, electronic signature data is not included regardless of this setting. Change the output device to view that data. This limitation does not apply to reports.

Archiving and Restoring Records

Use E-Signature Archive/Delete (36.12.14.22) to archive electronic signature records to a file and optionally delete the records from the system when they are no longer needed online.

If you need to access the records later, you can reload them using E-Signature Restore (36.12.14.23) based on ranges of signature dates and category codes. They are then available to E-Signature Report.

Select records by entering the last electronic signature creation date you want the system to consider. The system selects all records up to that date that have not previously been archived. The archive data file has the format of `esig_data_`*`YYYYMMDD`*`.hst`. If this archive function is executed multiple times a day, the data will be appended to the same file.

Note To ensure that signature-enabled programs can always display the latest signature data, the system does not delete the record for the latest signature event. It archives these records if they meet the selection criteria, but does not delete them even when Delete is Yes. The records are automatically deleted during a subsequent archive/delete session if they no longer represent the latest signature.

In E-Signature Restore (36.12.14.23), enter the date range of the records and the data file name to restore electronic signature records.

Chapter 7

Auditing

This section discusses how to set up the auditing functionality in your system.

Overview

Auditing is the process of evaluating an organization's practices for safeguarding electronic information from loss, damage, unintended disclosure, or denial of availability. It is one of the essential factors in providing a secure application and in meeting mandatory regulatory compliance.

The system's Auditing capability integrates with the Progress OpenEdge Auditing capability in Progress OpenEdge 10. Refer to the Progress documentation for additional information about the Auditing capability in Progress and about the Progress Data Administration utility.

With QAD's Auditing module, you can configure your system to maintain audit trails. Audit-trail records are created and stored in audit trail tables. They contain facts about changes made in the databases. A typical audit record includes information that helps you identify who made a change, which program made the change, when the change was made, and what the change was. You can set up these functions for all tables or you can limit the audit trail recording activity to specific tables.

The Auditing module adds value to the Progress OpenEdge Auditing capability by including:

- A user interface that allows you to enable and disable auditing at the table level with reusable defaults. This function is more straightforward than the audit policy maintenance function provided by Progress.
- A default audit policy. QAD's default policy includes configuration of identifying fields of tables in QAD's main database. With the default policy, users can easily identify changed records according to the content of the audit trail.
- Reports that perform better and are easier to use than the default Progress reports.
- The ability to trace the user who made the changes through the QAD architecture.
- Import/export policies to allow users to set up audit policy in one database and enforce it in other databases by exporting and importing the policy. This prevents having to repeat the setup for each database—reducing errors and ensuring that all company databases are using the same policy.

An Enhanced Controls license is required to use Auditing. Various OpenEdge utilities must also be run and data administration options set to enable particular databases for auditing.

Warning Importing policies of your own into the system or using Progress tools to change QAD's default policy may cause conflicts on policy configuration. These activities are not recommended.

Below is a list of menus and programs for the new auditing module.

Table 7.1
Auditing Module Menus and Programs

Number	Menu Label	Program
36.12.1	Audit Trail Report–App DB	atapprp.p
36.12.2	Audit Trail Report–Arc DB	atarcrp.p
36.12.13	Audit Trail Setup Menu	
36.12.13.1	Audit Policy Import	atplimp.p
36.12.13.2	Audit Policy Export	atplexp.p
36.12.13.5	Audit Configuration Maintenance	atplmt.p

Number	Menu Label	Program
36.12.13.6	Audit Configuration Report	atplrpt.p
36.12.13.11	Audit DB Maintenance	atdbmt.p
36.12.13.12	Audit DB Report	atdbrp.p

Planning Auditing

Thorough planning is necessary before setting up auditing and will save you a considerable amount of time. You should take into account certain system constraints when deciding which tables to enable for auditing. These planning considerations include:

- Which databases to audit
- Which tables to audit
- Using archive databases for reporting
- Auditing custom tables
- Schema changes

Determining Databases to Audit

Since most of the application data resides in the qaddb database, you should usually only audit-enable this database. However, you must also audit-enable the qadadmin database if you want to audit EDI tables such as the following:

- edtr_mstr
- edtrd_det
- edtrf_mstr
- edtrp_mstr
- edtrv_mstr
- edtxe_det
- edval_mstr
- edxf_mstr
- edxfd_det
- edxfdd_det
- edxfs_mstr
- edxfsd_det
- edxfsdd_det
- edxr_mstr
- edxrd_det
- edxref_mstr

Note The qadhelp database should not be audited since it only contains static system data.

Determining Tables to Audit

Most tables can be audited. However, some tables, such as those with a field type of raw or blob, cannot be audit-enabled due to technical limitations. For example, OpenEdge will generate a serious runtime error if you try to audit-enable any table with a field of the raw or blob data type. The system prevents you from enabling such tables for auditing, whether they are standard or custom tables. Audit Configuration Maintenance (36.12.13.5) will not show such tables and therefore cannot be audit-enabled. The same applies to Audit Configuration Report (36.12.13.6). As a result, you cannot see such tables in the report.

Note It is not necessary to audit tables of temporary usage within the system, such as qad_wkfl.

You might have to consider the performance overhead when deciding which tables to audit. The overhead depends on how many tables are audit-enabled and how often they change. As a rough guide, enabling auditing can cause the disk I/O for a table to increase two to three times. You usually only audit-enable those tables needed to meet your auditing requirements.

Before you audit-enable tables in a production database, QAD suggests that you validate your table audit selections in a test environment. This could include running simple tests to verify the data you need to audit is correctly collected and reported. This can eliminate unnecessary impact on your application if your table audit selections do not work as expected. If the test results are satisfactory, you can export the QAD main policy and load it into your production database.

Archive Database Considerations

You must set up the archive database to store the audit trail data to prevent the application database from growing unnecessarily large. Loading audit trail data into a dedicated archive database also has the advantage that it will not impact the performance of the application database when a lengthy audit report is generated from a separated database. You might consider distributing the audit trail data across a number of databases by date range. For example, you can create one archive database for each month, quarter or year, depending on the volume of audit trail generated in a particular period.

Auditing Custom Table Considerations

Auditing custom tables is just like auditing the standard tables—provided that the custom table schema follows QAD's convention. The primary keys should be meaningful data items so that they can serve as identifying fields for audit trail. Again, the only constraint is that you cannot audit-enable a table with any field of data type raw or blob.

Schema Change Considerations

If you want to make any major schema changes to an audit-enabled database (for example, updating to a new service pack), you must first disable auditing, make the change, and re-enable auditing for the database.

Note This is a workaround to a Progress bug. Applying a large incremental `.df` file to the audit-enabled database may result in error 13595. This issue has been submitted to Progress Software Corporation (PSC) with case number W804140215. It will be fixed by PSC in the OpenEdge 10.2A release.

Setting Up Auditing

To set up auditing, you must load the license for Enhanced Controls and then do the tasks described in the following sections:

- “Enabling Auditing for the Database” on page 115
- “Configuring Database Options and Audit Permissions” on page 116
- “Importing Audit Policy” on page 117
- “Enabling Auditing on Selected Tables” on page 119

Note When setting up auditing, QAD recommends that you first set up a test environment to test the auditing policy and then export the policy and load it into the production database.

Enabling Auditing for the Database

Before setting up auditing, do the following:

- Be sure that the database has been upgraded to Progress OpenEdge 10.1B03 or above.
- Back up the database.

To use auditing in Progress OpenEdge, you must add type II storage areas for the audit data and audit indexes to ensure optimal performance. The audit data and indexes can be in the same storage area, but QAD recommends that they be in separate areas. The descriptions in this document assume that you are using separate areas.

To add type II storage areas, use the `prostrct add` statement:

```
prostrct add <db> addaudit.st
```

Where `addaudit.st` contains the following information:

```
d "Audit_Data";64  . f 40960
d "Audit_Data";64  .
d "Audit_Index";64 . f 5120
d "Audit_Index";64 .
```

The above sample structure file uses a combination of fixed and variable length extents for maximum speed with a safety valve. You should set the limits to try to avoid hitting the variable length extent. The values used should be modified as appropriate for your system, but those shown represent a reasonable starting point. The sample code includes an example `adduadit.st` file.

Note The appropriate values for `addaudit.st` to use depend on many factors, including the size of your data, how much you expect to audit, the frequency of updates and throughput, and physical configuration.

Once you have added the type II storage areas to your databases, you can then enable auditing for each one as follows:

```
proutil <db-name> -C enableauditing area "Audit_Data" indexarea
"Audit_Index"
```

Note Databases can be individually audit-enabled. You are not required to enable all databases in a database set.

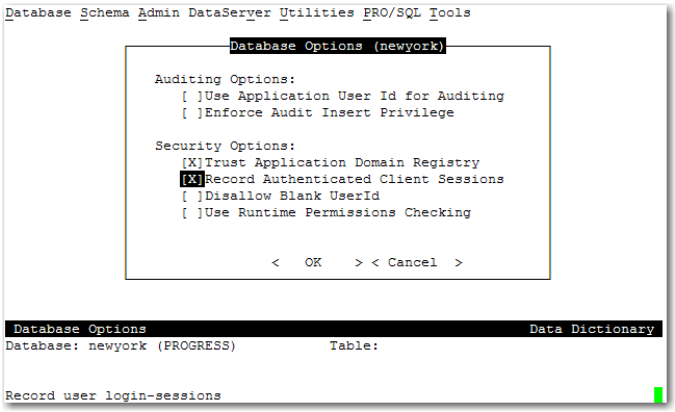
Configuring Database Options and Audit Permissions

For each of the databases you have updated and enabled for auditing, you need to configure the new database options and set up audit permissions.

Note To avoid the need to restart AppServers and the WebSpeed broker, you should log in to the database in single-user mode to perform any audit administration activities, such as defining database options. Otherwise, since the options are cached by the client for the current session, any changes you make will not take effect without restarting the system.

You set up the new database options in Progress using the Data Administration menu's Admin| Database Options settings.

Fig. 7.1
Database Options Settings



Select the following options:

- Trust Application Domain Registry
- Record Authenticated Client Sessions

To access audit-related menus, the user must be granted with specific audit permissions or roles. The roles are as follows:

- Audit Administrator — an authenticated user who has been granted privileges to create, update, and delete audit policies and read audit data.
- Audit Data Archiver — an authenticated user who has been granted privileges only to archive or load audit data. An audit data archiver has no access to the audit policy.
- Audit Data Reporter — an authenticated user who has been granted privileges to read the audit data.

Note The Audit Administrator is not an Audit Data Archiver or Audit Data Reporter. The Audit Administrator role does not include the other roles.

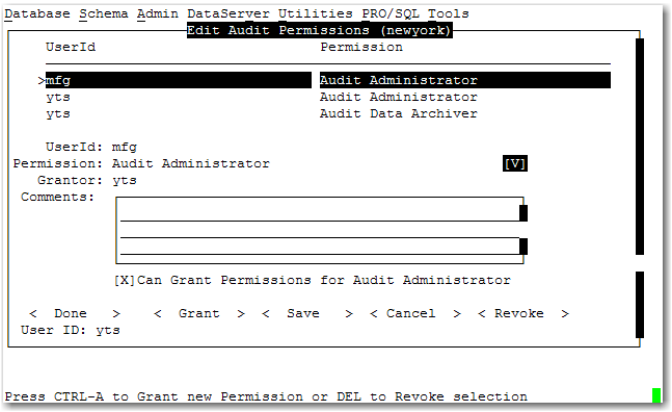
Table 7.2
Roles Required for Audit Menu Access

Program	Menu	Role Required
Enhanced Controls Menu	36.12	N/A
Audit Trail Report –App DB	36.12.1	Audit Data Reporter
Audit Trail Report–Arc DB	36.12.2	Audit Data Reporter

Program	Menu	Role Required
Audit Trail Setup Menu	36.12.13	N/A
Audit Policy Import	36.12.13.1	Audit Administrator
Audit Policy Export	36.12.13.2	Audit Administrator
Audit Configuration Maintenance	36.12.13.5	Audit Administrator
Audit Configuration Report	36.12.13.6	Audit Administrator
Audit DB Maintenance	36.12.13.11	None
Audit DB Report	36.12.13.12	None

To grant audit permissions, you can use the Progress Data Administration menu's Admin|Security| Edit Audit Permissions menu.

Fig. 7.2
Edit Audit Permissions Menu

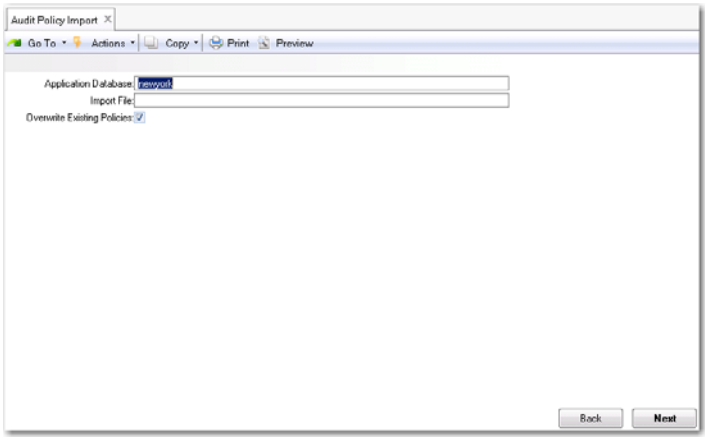


From this menu, you can reassign responsibility for the audit administration from the DBA to a separate audit administrator, and you can specify who can read and archive the audit data. For example, you might grant a particular user the Audit Administrator, Audit Data Archiver, and Audit Data Reporter permissions. (For further information, refer to the Progress documentation.)

Importing Audit Policy

You can import audit policies from Progress policy XML files using Audit Policy Import (36.12.13.1).

Fig. 7.3
Audit Policy Import (36.12.13.1)



Importing audit policies only needs to be done once for each audit-enabled database. The QAD solution will not work without the Progress default policy file loaded.

To import an audit policy file, open Audit Policy Import (36.12.13.1) and complete the following fields:

Application Database. Specify the logical name of the database where the policy will be applied.

Import File. Specify the location of the XML file containing the policy definition. Two files are normally loaded: a Progress policy file that defines general settings and the QAD policy file that defines settings relevant to application tables. The two files are `$DLC\auditing\policies.xml` (for the default settings of general activities such as login and changing schema, `$DLC` is the Progress installation directory) and `qadmainpolicy.xml` (default policy).

The reason you need a default policy (`qadmainpolicy.xml`) is as follows. When a table is audit enabled, an audit trail is created when a record is changed (an insert, modify, or delete). Important values in the audit trail include the values of identifying fields, which are used to identify the record. These values are displayed in an audit trail report and must be meaningful so that you can recognize a changed record without using database utilities. By default, Progress uses primary keys as identifying fields. If for a given table the primary key is not appropriate for identifying fields, identifying fields for the table need to be set. The default policy provides predefined identifying fields. When you use QAD's Auditing module, you should load the default policy so that you do not need to define the identifying fields.

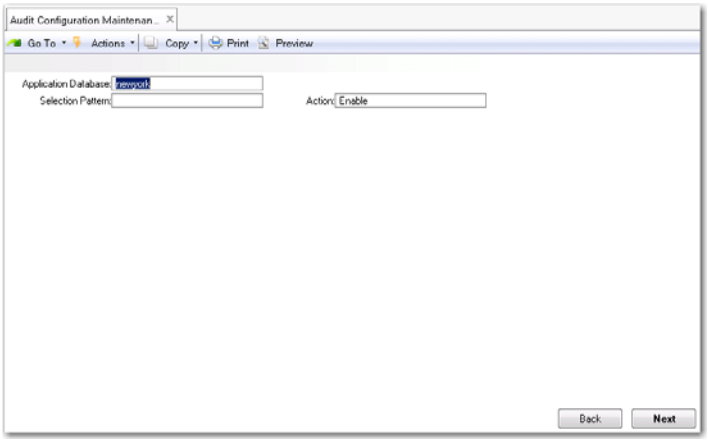
The default policy only covers tables whose identifying fields are different from primary keys. For most non-Financials tables, the primary keys can be used as identifying fields, so these tables do not appear in the default policy. For most Financials tables, however, the table has a primary key, which is a sequence identifier, and one or two unique indexes. One unique index is used for the identifying fields. Therefore, a Financials table and its identifying fields are included in the default policy.

Overwrite Existing Policies. Select Yes to replace current policies.

Enabling Auditing on Selected Tables

Use Audit Configuration Maintenance (36.12.13.5) to enable auditing on selected tables.

Fig. 7.4
Audit Configuration Maintenance (36.12.13.5)



To enable auditing on selected tables, first set the following fields:

Application Database. Specify the database name.

Selection Pattern. Leave this field blank to list all tables or enter a combination of letters and the asterisk symbol (*) as a wildcard to find tables based on matching. For example, to list only the tables whose names start with ab, enter `ab*`.

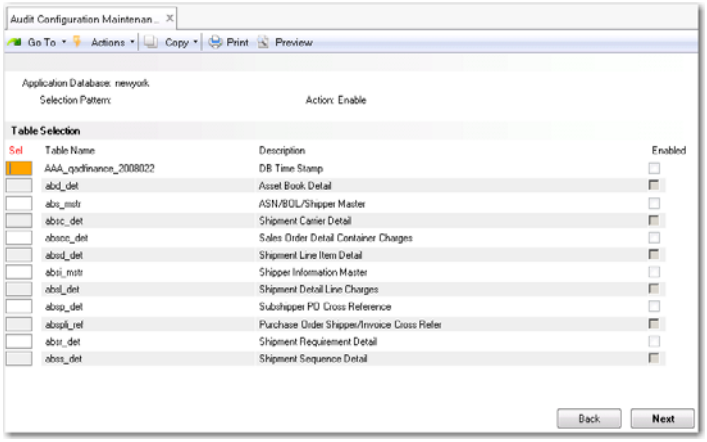
Action. Enter the code that represents the action you want to perform on selected tables:

Enable. Audit records will be created for selected tables.

Disable. Audit records will no longer be created for selected tables.

Click Next and the system lists all the tables that meet the selection pattern.

Fig. 7.5
Audit Configuration Maintenance Table Selection

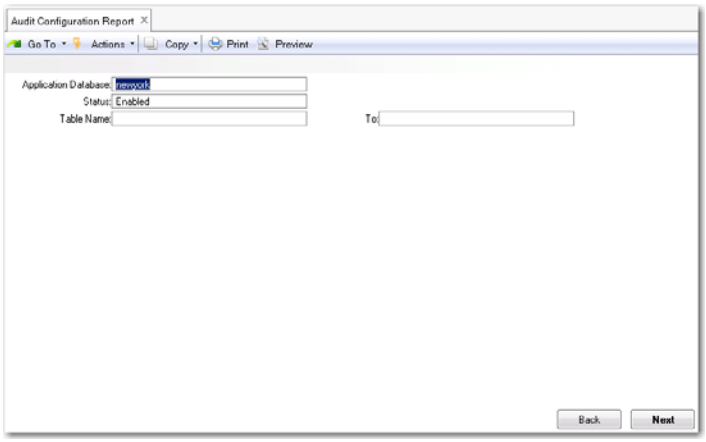


Use the up and down arrow keys to scroll through the list of tables. Use the space bar to select the tables whose Enabled settings you want to change. An asterisk symbol (*) displays at the front of the row of each table you select. To toggle the Enabled settings, select individual tables using the space bar and then press F1. The system asks whether you want to enable the selected tables. Enter Yes (or No) and then press the Enter key.

Generating Reports for Audit Configuration

You can use Audit Configuration Report (36.12.13.6) to get a report on what tables are audit enabled/disabled and their identifying fields.

Fig. 7.6
Audit Configuration Report



To generate an audit configuration report, specify the following:

Application Database. Specify the database name.

Status. Specify the auditing status of tables (Enabled, Disabled, or All).

Table Name and To. Specify a range of table names. Typically, you leave these fields blank to get a report of all tables that meet the Status criteria.

Press Go to have the system retrieve the records according to the criteria you have specified and generate a report (or output file).

For each table that meets the criteria, the report shows the name, whether the table has been enabled for auditing, and the identifying fields.

Setting Up Archive Database

QAD recommends that you create a separate archive database and archive the audit trail data to this database. There are several reasons to do this:

- Audit trail data requirements grow over time, requiring more disk space. Having a separate archive database prevents audit trail data from taking disk space in the application database.
- Running audit reports against the application database will impact system performance.
- The archive database can be deployed in a flexible way.

- The archive database can store audit trail data from different application databases and act as a central reporting database.

Creating Optional Archive Database

The optional archive database is a copy of the empty database with auditing enabled. No other schema needs to be loaded. This database is used only for reporting and is optional.

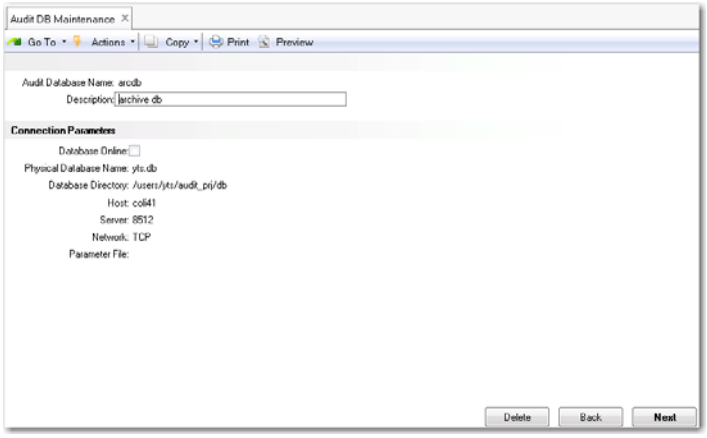
To create an optional archive database, do the following steps:

1. Create an empty database:
   `prodb <db-name> empty`
2. Enable auditing. For more information, see “Enabling Auditing for the Database” on page 115.
3. Log in to the database, create a user, and set privileges:
   `pro <db-name>`
   Access Progress Data Administration: choose Tools|Data Dictionary and press Enter. Next, choose the Admin|Security|Edit User List menu and create a user and password. Grant Audit Administrator and Audit Data Archiver privileges to this user. Save the settings and exit. (For further information, refer to the Progress documentation.)
4. Set up database options (see “Configuring Database Options and Audit Permissions” on page 116).
5. Use the template scripts in the QAD Enterprise Applications installation directory to load archived audit trail data.

Setting Archive Database Connection

Use Audit DB Maintenance (36.12.13.11) to create and maintain connection parameters for the archive databases.

Fig. 7.7
Audit DB Maintenance (36.12.13.11)



These connection parameters are used by Audit Trail Report–Arc DB (36.12.2).

This program is similar to Database Connection Maintenance (36.6.1), but has some important differences:

- You do not specify a logical name for the connection. The logical name is managed internally by the system.
- Connections to the archive databases are not permanent. The audit reports use the connection information to connect to the archive databases as needed. These processes do not maintain a connection to an archive database after they have retrieved the information they are handling.
- The system can connect to multiple archive databases simultaneously.

Important Archive databases must be configured and started up in multi-user mode before connecting to them using the connection parameters. Audit DB Maintenance does not start or stop archive databases. It only stores the connection parameters used to connect to them. You must set up external procedures to start up and shut down archive databases as needed.

Database connection parameters are defined by the way archive databases are implemented. The system administrator who creates and maintains the database provides the connection information required to set up the field values on this screen.

In the first frame, enter a name for this archive database connection record. The name must be 8 or fewer characters. It is used for tracking and maintaining your database connection information. It does not necessarily have to be the physical name given to the archive database.

Database Online. This field indicates whether the system should attempt to connect to the archive database. It does not indicate that an archive database is running, or that a connection to the database has been tested or is currently active.

Physical Database Name. Enter the actual physical name of the Progress database. Database names are typically case sensitive and can be up to 12 characters long. The database directory and physical name together make up the complete path name to this database. These are used on the database connect statement when connecting to this database.

Database Directory. Enter the complete path name of the operating system directory where this database is stored. Path names may be case sensitive and can be up to 50 characters long.

Host. Enter the name of the host server where the Progress database can be found. This name follows the `-H` parameter on the Progress connect statement. It is only required when the database is located on a different computer.

Server. Enter the name of the service to be used by the broker process when starting up the remote database. This name follows the `-S` parameter in the Progress connect statement. It is only required when the database is not located on the current machine.

Network. Enter the type of network being used. Valid values are TCP (default) and SNA (Progress/400). If left blank, TCP is assumed. This value follows the `-N` parameter on the Progress connect statement.

Parameter File. Specify the parameter file name. You must specify the connection parameters either in the parameter file or in the corresponding Host, Server, and Network fields. (Do not specify them in both.) If you use the parameter file:

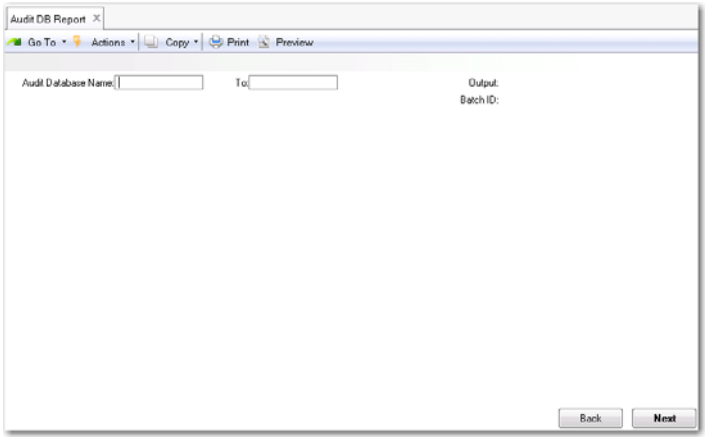
- The file must be accessible through the PROPATH or located in the directory specified in Database Directory.

- You must still specify the database name in Audit DB Maintenance (36.12.13.11). If the database is not located in the PROPATH, you must specify the full path in Database Directory.
- The file must not include the `-ld` or `-db` parameters.
- The file must include the `-trig` parameter, which specifies the location of trigger files.

Reporting Database Connection

You can use Audit DB Report (36.12.13.12) to get a report on the database connection and its parameters.

Fig. 7.8
Audit DB Report (36.12.13.12)



Enter the following:

Audit Database Name, To. Specify a range of database names. Typically, you leave these fields blank to get a report on the connection information of all archive databases.

Specify the output format and click Next to run the report.

Customizing Archive/Load Scripts

Four script templates are provided under the installation directory of QAD Enterprise Applications, two for UNIX (or Linux) and two for Windows:

UNIX:

`aud_archive.tpl` — Audit archive data

`aud_load.tpl` — Audit data load

Windows:

`aud_archive.wtp` — Audit data archive

`aud_load.wtp` — Audit data load

The basic approach is to allow the user to have the customized archive script running on the applications database server, and the load script running on the archive database server. Users can configure the scripts as batch tasks to automate the process as needed.

To use the templates, first choose the right templates according to the operating system type. Change the file extension to `.bat` for Windows scripts or `.sh` for UNIX (or Linux) scripts.

Next, modify the related settings:

- Set the database name, database directory, and DLC.
- Make sure the Progress database user has a non-blank password and proper privileges (Audit Data Archiver).

Run the archive script. The archive file named *`DBNAME`*`.abd` is generated in the directory *`ARCHDIR/DBNAME/DATEDIR`*, where *`DATEDIR`* is named based on the archive date and time. The related information is recorded in the log file in the *`LOG`* directory. The user is informed of the result. Check the log file if there are any issues.

Run the load script. In some circumstances, before loading the data, you might need to manually move the file to a location that the load script can access.

The UNIX (or Linux) scripts can also send e-mail to the mail address *`USER_TO_NOTIFY`* to report the result.

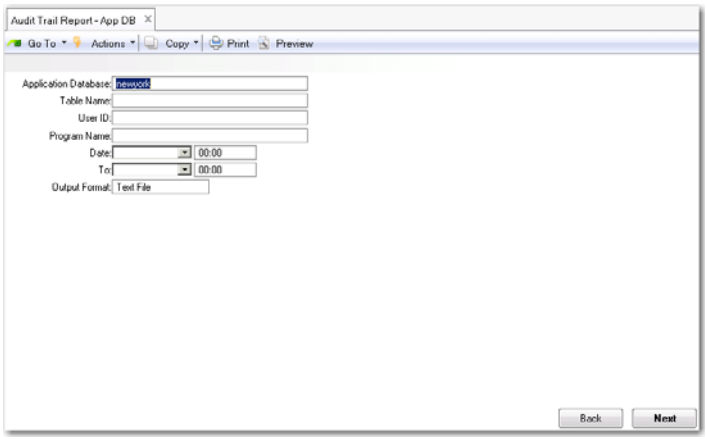
Generating Audit Trail Reports

You can generate reports against application databases and against archive databases.

Generating Reports Against Application Databases

You can generate auditing reports against the application databases using Audit Trail Report–App DB (36.12.1).

Fig. 7.9
Audit Trail Report–App DB (36.12.1)



To generate audit reports, specify the following:

Application Database. Select one of the connected databases. If the database selected is not enabled for auditing, the system displays an error message.

Table Name. Specify a table, or leave this field blank to specify all tables.

User ID. Specify a user ID, or leave this field blank to specify all users.

Program Name. Specify the name of the program that changed the records. For example, specify `sosomt.p` (Sales Order Maintenance) to get audit trail data generated for changes made by Sales Order Maintenance.

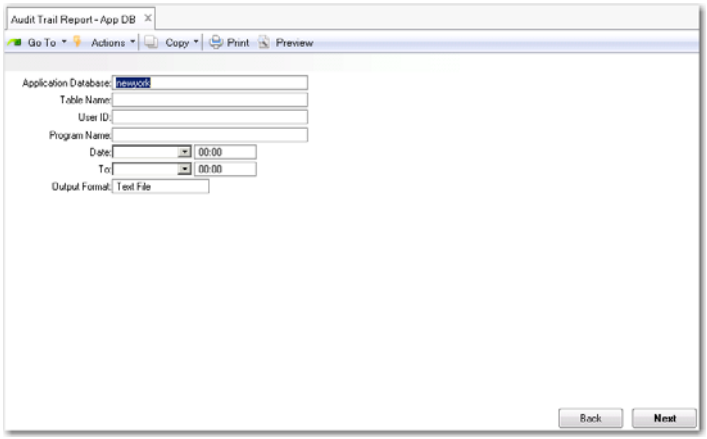
Date, To. Specify the beginning and ending dates.

Summary/Detail. Select summary mode or detail mode. In summary mode, no field value changes are reported.

Output Format. Specify the output format as a text file (`.txt`) or XML file (`.xml`).

Click Next. If just one table was selected, then the system displays the identifying fields and allows the user to specify a range of values for one or more fields to narrow the search results. This frame displays the field name and the field label for each identifying field of the selected database table. You can specify from and to values in the data range frame.

Fig. 7.10
Audit Trail Report–App DB Field Selection



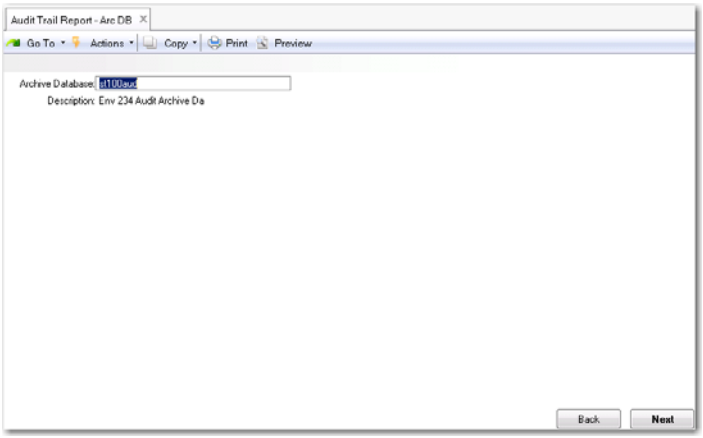
Click Next again to have the system retrieve the records according to the criteria you have specified and generate a report (or output file).

The report shows the Table Name, User ID, Date, Time, and Program for each audit trail record that meets the criteria.

Generating Reports from Archive Databases

You can generate auditing reports from the archive databases using Audit Trail Report–Arc DB (36.12.2).

Fig. 7.11
Audit Trail Report–Arc DB (36.12.2)



To generate audit reports, specify the following:

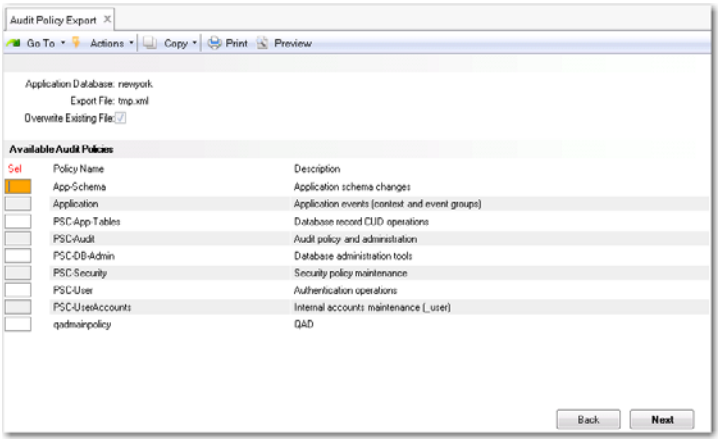
Archive Database. Select one of the archive databases. If the database selected is not enabled for auditing, the system displays an error message. Use Audit DB Maintenance (36.12.13.11) to define archive database information.

Click Next to have the system retrieve the records according to the criteria you have specified and generate a report (or output file).

Exporting Audit Policy

You can export current audit policies to Progress policy XML files using Audit Policy Export (36.12.13.2).

Fig. 7.12
Audit Policy Export (36.12.13.2)



To export to an audit policy file, specify the following:

Application Database. Select one of the applications databases. (If the database selected is not enabled for auditing, you will get an error message.)

Export File. Specify the name of the Progress policy XML file to which you want to export the audit policy.

Overwrite Existing File. Specify whether you want to overwrite an existing Progress policy XML file.

Click Next and the system lists all the current audit policies. Use the up and down arrow keys to scroll through the list of policies. Use the space bar to select the policies you want to export. An asterisk symbol (*) displays at the front of the row of each policy you select. Click Next (or in the Character UI, press F1) and the system prompts you to export the selected policies to the export file you have specified.

Disabling Auditing

As an Audit Administrator, you can disable auditing for a database, but the database must be offline when you disable auditing for that database. Disabling auditing does not remove any recorded audit data or the auditing tables. Access to the audit data remains restricted to authorized users when auditing is disabled.

You can disable auditing for a database using the following command:

```
proutil <db-name> -C disableauditing
```

Index