QAD Enterprise Applications
Standard Edition

User Guide
Manager Functions

Setting Up and Using Domains
Domain Constants
System Interface
Printers and Batch Processing
CIM Interface
Database Management
Reports and Utilities
System Cross-Reference
Application Server
User Interface Management
Users and Security
Electronic Signatures
Audit Trails
Domain Reference

78-0872A
QAD Enterprise Applications 2010
Standard Edition
June 2010

ManagerFunctions_UG_v2010SE.pdf/mat/mat

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

Contents

Chapter 1

Introduction to Manager Functions

Manager Functions includes tasks typically performed by system administrators. Most functions located on the Manager Functions menu (36) are discussed in this guide.

A few functions on the Manager Functions menu are discussed in other guides:

- Domain/Account Control (36.1) affects processes throughout the system. However, it is not typically set up by system administrators, but by individuals in your company with financial expertise. It is discussed in *User Guide: Financials*.
- Configured Messaging (36.4) applies only to scheduled orders and is discussed in *User Guide: Release Management.*
- External Interfaces (36.5), Q/LinQ (36.8), and the Logistics API (36.5.7) are discussed in the user guides for various add-on products.

This guide does not cover the various utilities on the Manager Functions menus numbered above 24. For documentation of these programs, see the procedure help or the opening program screen of each utility.

Areas covered in this guide are described briefly below.

Setting Up and Using Domains 5

The domain concept in the database provides flexible implementation options for supporting multiple business operations within a single database and eliminates the need for a single database-wide base currency or database-wide control settings. The domain is essentially a logical partition within a single database. Any number of domains can be set up in one physical database—each domain with its own base currency, chart of accounts, operating controls, document numbering, and security.

Domain Constants 23

The programs on the Domain Constants menu (36.2) control calendars and codes used throughout an individual domain. These include shop and holiday calendars, reason and generalized codes, and rounding methods.

In addition, you can set up number sequences using number range management (NRM) functions, which support regulatory controlled document numbering. NRM includes the content and sequencing of a numeric series, as well as preventing gaps in a series.Finally, you can specify fields in tables for detailed change tracking and reporting.

System Interface 43
The System Interface menu contains programs that control menus, screen labels, messages, multi-language installations, and help. You can also set up user function keys and define your e-mail system.

Printers and Batch Processing 55
The Printer Management menu contains programs for setting up system printers, specifying default printers by user or group, and creating batch print requests.

CIM Interface 65
CIM (computer integrated manufacturing) is one way to load legacy or non-Progress data into the database. Using CIM, data can be added using standard program validation.

Database Management 75
The system provides utilities for monitoring database size, performing dumps and loads, reloading archive files, and managing database sequences. Delete/archive followed by dump/load is the standard means of controlling database size and fragmentation in Progress databases. User licensing utilities and programs for managing time zones are also included in database management.

Reports and Utilities 99
A number of system-wide reports and utilities are provided on the Manager Functions menu.

System Cross-Reference 103
The system cross-reference programs display information about field, program, and table relationships in your database. If you customize the product, this is an essential set of programs.

Application Server 109
The system can use a Progress application server (AppServer) to run applications remotely. The AppServer must be defined to make it available.

User Interface Management 117
The UI: Manager Functions menu provides programs used to create browses and associate them with fields and programs. You can also define alternate programs to execute when menu items are selected and specify programs to be run from other programs.

Users and Security 133
A user must be defined with a valid ID and password before they can log in. In addition, the system offers several types of security, including domain, menu, field, entity, site, account, and inventory movement code. You can implement these levels by user ID or user group.

Enhanced Controls

The optional Enhanced Controls module allows companies to track in detail who made changes to enterprise-critical data, what changes were made, and when. Components of the Enhanced Controls module include:

Electronic Signatures 177
Use the electronic signature functions of the Enhanced Controls module to apply electronic signature requirements to a subset of menu programs and database tables you choose from QAD-provided default setup data. Signature records include data such as:

- Identification of the user who created or modified the data

- An indication of whether the data has been updated since it was most recently signed
- Remarks the user entered when signing the data
- Detailed field values for all elements of the signature record

Audit Trails 209

Use the audit trail functions to maintain multiple separate audit databases containing a history of changes made to records associated with the database tables you choose. Audit trail records include data such as:

- Identifying information for the user performing the update
- Date, time, and time zone when the change was posted
- Before and after data values for changes, additions, and deletions

Chapter 2

Setting Up and Using Domains

This chapter describes the concept of domains and how to set them up and use them. It also includes information on using domain features across databases and across domains within a single database.

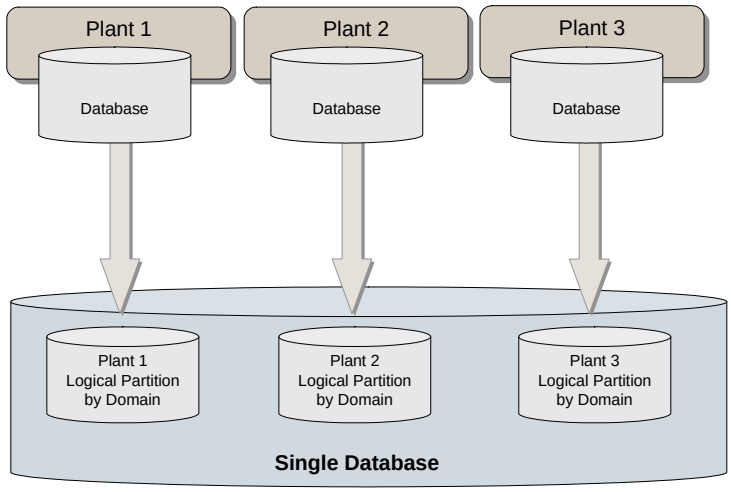
Introduction to Shared-Services Domain

Note Shared Services Domain is a separately licensed module. Unless you purchase appropriate licences, the system prevents you from having more than one active domain per database.

The domain concept in the database provides flexible implementation options for supporting multiple business operations within a single database and eliminates the need for a single database-wide base currency or database-wide control settings. The domain is essentially a logical partition within a single database. Any number of domains can be set up in one physical database—each domain with its own base currency, chart of accounts, operating controls, document numbering, and security.

Figure 2.1 illustrates how multiple databases can be mapped into multiple domains within one database.

Fig. 2.1
Domain Solution



Note A database with one or multiple domains can continue to connect to another database also with one or multiple domains.

Some system administration functions can be managed across domains, such as defining users, currency codes, country codes, menus, messages, and labels. This includes the ability for a system administrator to control exactly which users can access data in which domains. All other data updates take place within the context of a specific domain.

Replication tools let system administrators synchronize common master data across domains, where appropriate. Processes that currently operate between databases can be used between domains within a database in a more streamlined and reliable manner. These processes include distribution requirements planning (DRP), enterprise material transfer (EMT), and enterprise operations planning.

Combining operations in a single database can have the following advantages:

- Facilitate the standardization of business processes among operational units.
- Reduce IS costs as a result of having fewer databases to manage.
- Facilitate reporting and custom queries because all data is stored in the same table structure referenced by the domain field.

- Support data sharing.

Each business can choose the most appropriate deployment method:

- A solution with a different database for each business operation
- A central solution with one database serving all
- Any combination of these

Domain Setup Overview

As part of any initial application implementation, you must perform a number of setup tasks including the following:

- Setting up system-wide data such as printers, menus, messages, and language codes
- Defining users and security
- Setting up financial data such as your chart of accounts and entities
- Defining master data such as items, sites, customers and suppliers

You must also complete tasks related to setting up domains. This section highlights setup activities related to domains.

Implementing Domains

Table 15.1, “Non-Domained Tables,” on page 238 lists the tables in the database that contain data that applies to the entire database. Data in all other tables is specific to a particular domain. You should consider the implications for data setup carefully during implementation to ensure that users who can change domains do not encounter validation errors.

Generalized Codes Example

Generalized codes are domain specific. This is because when domains represent businesses in diverse geographical and political locations, these codes may vary widely. For example, customer types, sales distribution channels, and buyer/planner codes could differ between a domain representing a business in England and one in Germany.

However, some programs that update system-wide data such as User Maintenance (36.3.1) also reference generalized codes. These generalized codes must exist in all domains or you may encounter errors editing a user record in one domain that do not occur in another.

Streamlining Setup

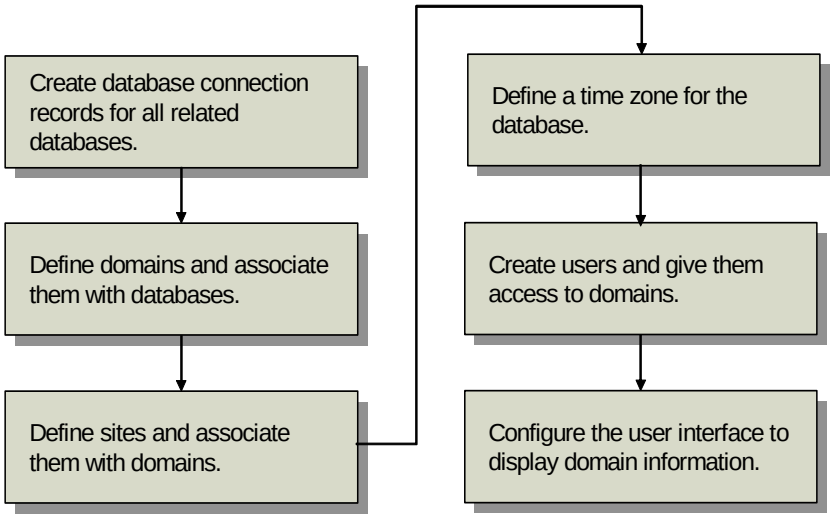
If you have several domains with similar base data, you can use alternate approaches to streamlining data setup:

- You can set up base data in the system domain. When you create a new domain, the system domain is used as a template and the new domain automatically inherits the same system data. See “System Domain” on page 9.
- You can use the synchronization features of Q/LinQ to replicate data from selected tables across domains. See Chapter 16, “Using Q/LinQ with Multiple Domains,” on page 245.

Domain Setup Work Flow

Figure 2.2 illustrates the steps required to set up domains in a database.

Fig. 2.2
Domain Setup Work Flow



1. Identify each database in Database Connection Maintenance (36.6.1) and define its location and connection parameters. You must create a record for the working database even when you are not using multiple databases. See page 8.
2. Create domains in Domain Maintenance (36.10.1) and associate them with databases. Domain names should be unique across connected databases. See page 9.
3. Assign each site to the appropriate domain in Site Maintenance (1.1.13). See page 14.
4. Define a time zone for the database in Database Control (36.24). See page 14.
5. Create users and give them access to domains in User Maintenance (36.3.1). See page 15.
6. Configure settings that affect the display of domain information on the user interface using Security Control (36.24) and Program Information Maintenance (36.3.21.1). See page 15.

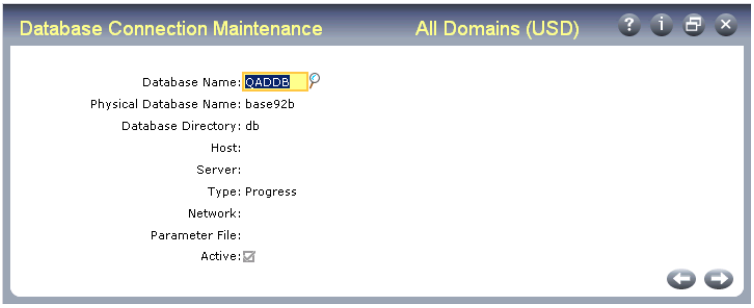
Creating Database Records

In a multiple-database environment, use Database Connection Maintenance (36.6.1) to specify the databases on your network and how to connect to them.

For a single database, you must still create a connection record that defines the database name. Only databases defined in this program can be associated with domains in Domain Maintenance.

Note A record for qaddb is automatically loaded with the system data during installation.

Fig. 2.3
Database Connection Maintenance (36.6.1)



Creating Domains

Use Domain Maintenance (36.10.1) to define domains in the current database. You create two types of domains:

- *Primary* domains reference the current database.
- *Connection* domains point to domains located in other databases.

Most functions update data within a specific domain. For example, each domain has its own base currency, chart of accounts, and control settings. All business documents—such as sales orders, purchase orders, and work orders—reference a specific domain.

A few system maintenance functions update data shared by all domains. These include functions such as printers, users, menus, messages, and currency codes.

Table 15.2 on page 240 lists these functions.

The Header Display Mode setting in Security Control (36.3.24) determines if the current domain name displays in program title bars in the character and Windows user interfaces. When the domain name displays, programs that update shared data display All Domains in the title bar. Programs that update domain-specific data display the domain short name and currency instead.

See “Setting Display Mode” on page 15.

System Domain

Every database must have one system domain, indicated by a domain type of SYSTEM. The initial system domain is created when the database is created, for both a new installation or a conversion. The initial system domain code is QAD. You can change the domain name and short name—but not the domain code—using Domain Maintenance as needed.

The system domain includes default data that is required to begin implementation, such as control program settings, rounding methods, default accounts, and generalized codes.

See Table 15.3, “Tables Copied for New Domain,” on page 243.

The system domain is used as a template for new domains. When you create a new domain associated with the current database, default data is copied from the system domain. This default data is not added to connection records, which reference another database that contains the actual data associated with a domain.

Since the system domain is used as a template, you may want to add data to it or tailor defaults before creating new domains based on it.

The system domain is typically not used for maintaining active transactions. You can prevent users from updating it by setting its Active field to No and by restricting access in User Maintenance (36.3.1).

See “Giving Users Access to Domains” on page 15.

Multiple Database Validations

When you create a domain, you must associate it with a database. When you create a new primary domain (database is your current database) in a multi-database environment, all databases must be connected. The system verifies that the domain you are about to create does not already exist as a primary domain in another connected database. If it does, an error displays and you cannot continue.

If the database you specify is not the database you are currently logged in to, the domain is considered a connection record. Normally, you do not need to create connection records manually. When you create a new primary domain, the system automatically creates connection records in other databases defined in Database Connection Maintenance (36.6.1).

If you do create a connection record manually, the system verifies that the domain exists as a primary domain in a connected database. Otherwise, an error is generated.

When you are using multiple databases operating over a network, the system uses the domain associated with a site to determine where database records should be updated.

Active and Inactive Domains

To ensure data integrity, you cannot delete a domain. Instead, set the Active field to No. This prevents users from specifying this domain at log-in or using Change Current Domain (36.10.13) to switch to it later.

Note Unless you have a license for Shared Services Domain, only one domain can be active at the same time.

In a multiple-database environment, you can only change the active status of domains in the current database, and then only when all other databases are active and connected. The system modifies the Active field for the connection records that exist in the other databases. An error displays if any database cannot be accessed and you cannot change the active status.

Domain PROPATH

When you use domains to combine multiple business operations in one database, each domain may require unique product licensing agreements or localizations.

When the database is started, the Progress PROPATH environment variable sets the directory paths that the system uses to locate and run Progress executable programs. Values set in the PROPATH can point to different directories for different sets of programs, or multiple versions of the same set

of programs. Use the Propath Setting field in Domain Maintenance to associate each domain with a specific set of PROPATH entries—so that the system automatically runs the correct program code for the current domain.

Note Typically, setting additional values in Propath is needed only under very specific circumstances. Most system administrators can leave it set to the default value, blank.

When you log in to a domain or switch to a domain that has a value entered in Propath Setting, the system updates your current default PROPATH by adding the domain-specific directories to the front. This allows domain-specific programs to be found before those in your default PROPATH, which is assigned at login. Each time you switch domains, the system clears any PROPATH changes made for the previous domain and adds any values specified in the Propath field for the new domain.

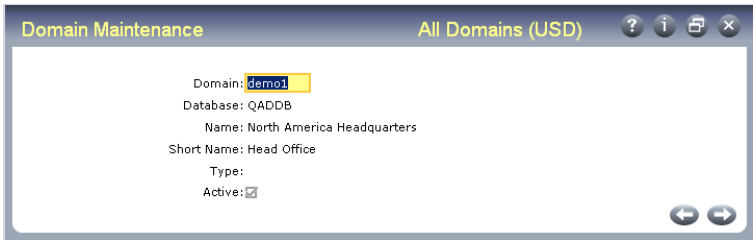
Use this setting if you have custom or localized code that applies to the business requirements of only a particular domain.

Example Your database has two domains: one for operations in the US and one for operations in Brazil. Shipments made in the Brazil domain require specialized documents that are generated based on a modification to one of the shipment programs. Specify the path to this localized program in the Brazil domain Propath field.

Domain Maintenance

Figure 2.4 illustrates Domain Maintenance (36.10.1).

Fig. 2.4
Domain Maintenance (36.10.1)



Domain Code. Enter a code (up to 8 characters) identifying a specific domain. Codes are restricted to the characters A–Z, a–z, and 0–9.

Domain Name. Enter a descriptive name to associate with this domain (up to 28 characters). This name must be unique within a database and across connected databases.

This name displays in the lookup associated with domain fields and on various reports and inquiries, as space permits.

Note When you change the domain name, the system automatically sets the value of ~SCREENS and ~REPORTS to the new domain name. You define these values in Company Address Maintenance (2.12) to represent your company name on the top of menus and reports.

Domain Short Name. Enter a brief name (up to 14 characters) to associate with this domain. This name must be unique within a database and across connected databases.

The domain short name displays in the program title bar in the character and Windows interfaces based on the setting of Header Display Mode in Security Control. It always displays in the program title in Desktop screens.

See “Setting Display Mode” on page 15.

Database. For a new domain, enter the name of the database where the domain is located. Set up databases in Database Connection Maintenance (36.6.1). You must specify a database even in a single-database environment.This is to ensure proper setup data exists if you decide to add other databases later.

Once a domain exists, this field cannot be edited. Database defaults to your current working database.

Domain Type. Enter a code identifying the type of domain. You can use this field to group domains based on a user-defined convention.

One domain in each database must be defined with a domain type of SYSTEM, which is used as a template for supplying default data when other domains are created.

You cannot modify the type of the system domain. However, you can change another domain to be the system domain by modifying its type to SYSTEM. In this case, you are prompted to continue. If you respond Yes, the type of the current system domain is set to blank and the domain you are editing becomes the system domain.

See “System Domain” on page 9.

Note A connection record cannot have a type of SYSTEM.

Active. Indicate whether this primary domain is currently active.

Yes (the default): This domain can be associated with users in User Maintenance and specified at log-in.

No: This domain is not active in the current database.

Note Unless you have purchased the Shared Services Domain module, the system lets you have only one active domain. If you attempt to activate a domain when your database already has an active domain, an error message displays.

When new sites are created in Site Maintenance (1.1.13), a site connection record is created in active domains only.

See “Associating Domains with Sites” on page 14.

The system performs the following validations related to this field:

- You cannot change the Active setting of your current domain. You must switch domains first and then modify the other domain to inactive.
- You cannot change this field if the domain is a connection record (referencing another database). You must change this field in the domain’s primary database.
- In a multiple-database environment, you can only change this field for a domain in the current database when all other databases are active and connected. The system modifies the Active field for the connection records that exist in the other databases. An error displays if any database cannot be accessed and the field cannot be changed.

Propath Setting. Enter a comma-separated list of directories—in addition to those defined at login—that the system should use for this domain. The system validates that the entry does not exceed 160 characters—the maximum size of the database field—and that all elements represent valid directories.

You cannot change this field for the current working domain; in this case, the Propath field is disabled. If necessary, switch to a different domain.

See “Domain PROPATH” on page 10.

Important You can only update this field in the character and Windows user interfaces. However, this does not limit QAD Desktop and .NET UI from running domain-specific programs based on the values in the field.

Viewing Domain Information

Use either of the following two functions to view information about domains:

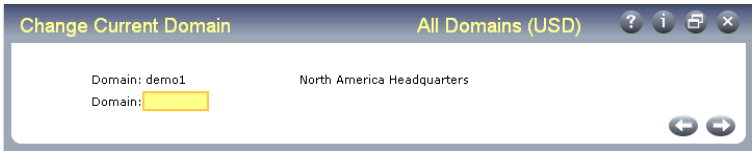
- Domain Browse (36.10.2)
- Domain Report (36.10.3)

Changing the Current Domain

You can use Change Current Domain (36.10.13) to change the active domain in your current session to another domain associated with your ID in User Maintenance (36.3.1).

Figure 2.5 illustrates Change Current Domain.

Fig. 2.5
Change Current Domain (36.10.13)



Note If only one domain is assigned to you, an error displays when you attempt to execute this program.

This function is useful for system administrators, corporate controllers, or others with system-wide responsibility who regularly access and update information in multiple domains.

This function affects your current session only. Each time you log in, you are prompted to specify a domain. The domain designated as default in User Maintenance displays by default.

When you change domains, the system accesses information about the new domain such as the base currency and primary entity.

Note Changing domains does not affect the domain associated with detached windows in QAD Desktop or .NET UI.

Domain Access

You can only change to an active domain you have been given access to in User Maintenance. If you are assigned to a different user group in the new domain, the functions you can perform may be different from the functions you performed in the previous domain.

Database Switching

If you change to a domain associated with a database other than the current one, database switching is initiated. The system connects to the database using the information set up in Database Connection Maintenance. If the connection cannot be made, a message displays.

This is equivalent to logging out of the system and starting a new session in a different database.

Note When you switch databases using this program, the system checks your security access based on the user groups defined for your user ID in the target domain and database.

Associating Domains with Sites

When you define a new site in Site Maintenance (1.1.13), you must associate it with a domain. When a site is associated with the current working domain, it is considered a primary site; otherwise, it is a connection record pointing to the domain where the actual site-related data is maintained. The other domain can be in this database or in another connected database.

When you are using multiple databases operating over a network, the system uses the domain associated with a site to determine where database records should be updated.

To help you manage the relationships among sites and domains, you can optionally create connection records in related domains when you create a new primary site.

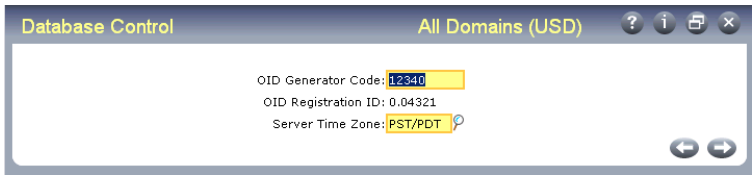
See *User Guide: Master Data* for information on sites.

Setting a Default Time Zone

The server time zone applies to the entire database. Use Database Control (36.24) to specify the time zone of the database. You should do this before defining users since the time zone specified here defaults when new user records are created.

See “Setting Up Multiple Time Zones” on page 94.

Fig. 2.6
Database Control (36.24)



Enter the time zone associated with the server machine for the current database. The system verifies that this is a valid time zone defined in Multiple Time Zones Maintenance (36.16.22.1).

When a new user is created in User Maintenance (36.3.1), the user time zone defaults from the server time zone.

If you are using the optional Service/Support Management module and the Multiple Time Zone option is activated in Service Management Control (11.24) for any domain in the database, this field cannot be modified here. Instead, you must use the Server Time Zone Change Utility (11.21.22.22).

Note The OID Generator Code in Database Control is used to assign unique object identifiers (OIDs) to database records for auditing purposes. The code is assigned during system implementation. See the installation guide for your system for information.

Giving Users Access to Domains

Use User Maintenance (36.3.1) to create users and assign each user access to one or more domains. You can:

- Specify one or more domains in the current database that this user can access. Menu functions the user can execute in each domain are determined by the user's group assignment.
- For users who can execute functions in more than one domain, indicate which domain they normally use. This domain is the default during log-in.
- Assign the user to one or more groups in each domain this user can access. Use groups to streamline security setup for menus, entities, sites, and other functions that allow specification of a user group. This is an optional feature. See "Controlling Access with User Groups" on page 163.

See "Specifying Domains" on page 161.

Note 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.

Configuring UI Settings

Domains affect a user's working context. In a multiple-domain environment, users need to know their current working domain and currency, as well as when they are using a program that updates information that applies to all domains.

This section discusses some of the ways you can configure and view domain-related information on the UI:

- You can use settings in Security Control (36.3.24) and Program Information Maintenance (36.3.21.1) to control the information about domains that displays on each screen.
- You can view the user's working domain in Session Master Maintenance (36.20.10.15).
- You can review context information using the Ctrl+F key combination.

Note Some of these settings affect the character and Windows UIs only.

Setting Display Mode

Use the Header Display Mode field in Security Control (36.3.24) to control the information that displays in the menu and program title bars of programs in the character and Windows user interfaces.

Note Display mode does not affect the display of programs in QAD Desktop or the .NET UI. For information on those user interfaces, see the user guide for the appropriate UI.

Based on such factors as security requirements, you can choose to display:

The date only

The ID of the logged-in user only

The date and domain name

The user ID and domain name

See page 150 for details about each display mode.

When you select an option that includes the domain name, the name of the program currently being executed no longer displays on the UI. In the character and Windows interfaces, you can use the Ctrl+F key combination to review this information and other context details.

See “Using Ctrl+F to View Information” on page 17.

Updating Program Information

You can use a setting in Program Information Maintenance (36.3.21.1) to control which programs display the string All Domains in the title bar.

- In the character and Windows interfaces, this displays when Header Display Mode is 2 or 3.
- In Desktop, either the domain name or All Domains displays regardless of the Security Control setting.

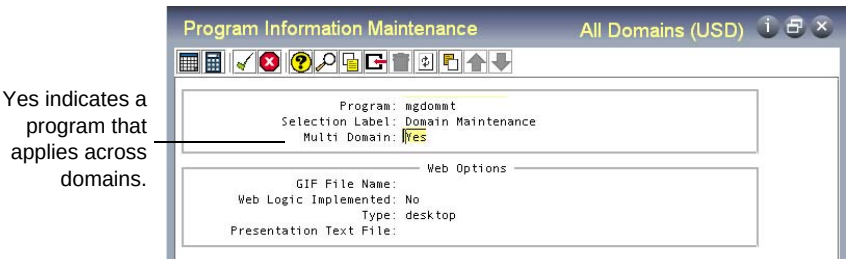
Note This field affects all UIs.

Information about all programs is initially loaded into your database during installation with appropriate default settings. You can update the setting for your custom programs or change it if you want the current working domain to continue to display even when a user is updating a table that applies across domains.

See Table 15.2 on page 240 for a list of programs that display All Domains.

Note This change affects what displays on the UI, only. The program continues to update data for all domains.

Fig. 2.7
Program Information Maintenance (36.3.21.1)



Multi Domain. Indicate if this program updates data that applies to all domains in the database.

No: The data referenced by this program is specific to the current working domain. For example, generalized codes apply to each domain separately so Multi Domain is set to No by default for `mgcodemt.p`.

Yes: The data referenced by this program is not part of a specific domain. For example, country codes apply to the database as a whole so by default Multi Domain is set to Yes by default for `adctrymt.p`.

When Header Display Mode is 2 or 3 in Security Control and a user invokes a program, the system checks the value of this setting to determine what to display in the program title bar.

- When Multi Domain is No, the short name and base currency of the current working domain display.
- When Multi Domain is Yes, the string All Domains displays in the header to help users easily identify functions that operate across domains.

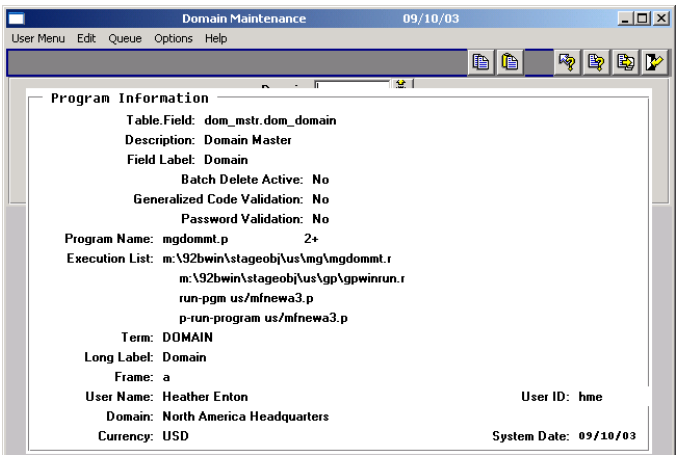
Note Base currency of the working domain continues to display.

Using Ctrl+F to View Information

Ctrl+F displays a pop-up window like the one illustrated in Figure 2.8 with more complete information about the context of the current field. This includes the program name being executed.

Note Ctrl+F works in character and GUI interfaces only.

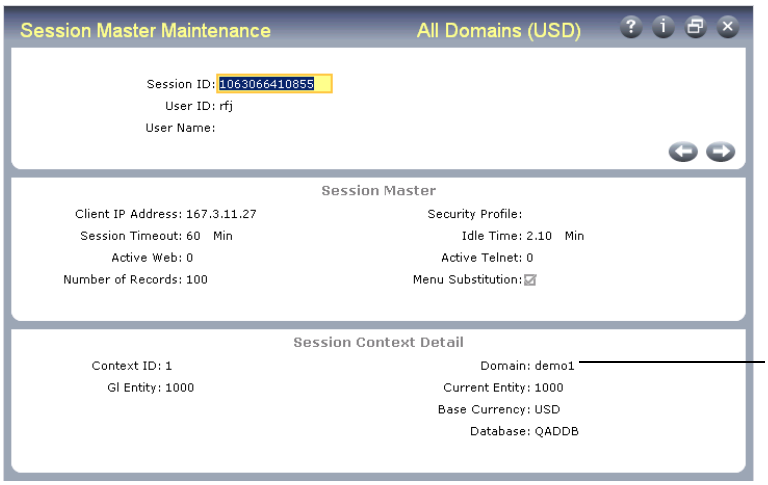
Fig. 2.8
Ctrl+F Pop-Up Display



Viewing Session Details

You can use Session Master Maintenance (36.20.10.15) to view information about users currently logged in to the system and details about their log-in sessions, including the current working domain.

Fig. 2.9
Session Master Maintenance (36.20.10.15)



Using Cross-Domain Features

Some functions update data in remote, connected databases. These functions can also be used to update data in domains within the same database. In addition, a number of functions provide visibility across domains in a database. This section discusses these two types of functions.

Using Multi-Database Functions Across Domains

A number of functions update data in more than one domain or database. Database switching is automatically initiated when the domain associated with one of the sites referenced in the function exists in a different database. The system determines the database involved based on information defined in Site Maintenance (1.1.13).

This section discusses three such functions and points out the differences in how they work across domains rather than across databases.

- Enterprise Material Transfer (EMT)
- Distribution Requirements Planning (DRP)
- Enterprise Operations Planning (EOP)

Using Enterprise Material Transfer

When Use Enterprise Material Transfer is Yes in Sales Order Control (7.1.24), you can create EMT sales orders. EMT automatically translates sales orders into purchase orders. You then transmit them to the appropriate supplier using EDI ECommerce. EMT also manages and coordinates changes so that sales order and purchase order information is synchronized.

See *User Guide: Distribution* for details.

Multi-level EMT manages orders across multiple levels within an organization. Order changes can be made at the top or bottom of the hierarchy and are then transmitted up or down to the next level.

EMT can function between business organizations in one domain, in different domains within the same database, or in different databases. The way EMT functions in these various scenarios is basically the same. However, if the related business units are represented by different domains within the same database, it becomes easier to use the direct allocation feature of EMT.

With direct allocation, the primary business unit (PBU) can make a special, temporary allocation of an EMT sales order or material order line item at the secondary business unit (SBU) site. When the SBU imports the PBU's EMT purchase order to create a secondary sales order, the system automatically converts this temporary allocation to a general allocation.

When the different business units in an EMT relationship are in separate databases, direct allocation can be used only when the business partners use the same version of QAD Enterprise Applications and when the databases are connected.

The following list summarizes the scenarios supported for EMT:

- Within a single database where the PBU and SBU are within the same domain.
- Within a single database where the PBU and SBU are in different domains.
- Across multiple databases where the PBU and SBU are in different domains.
- Across multiple databases where the PBU is in a database with domains (later versions of QAD's ERP application) and the SBU is in a database on an earlier release. In this case, direct allocation cannot be used.
- Across multiple databases where the PBU is in a database without domains and the SBU is in a database with domains. In this case, direct allocation cannot be used.

Note The SBU could also be using a non-QAD system. Direct allocation is not supported in this scenario either, regardless of the PBU's application version.

Using Distribution Requirements Planning

In a multiple database environment, you can use DRP to plan supply to meet demand for multiple sites within the current database and distribute demand to other connected databases. DRP can be used only under the following conditions:

- The base currencies of the databases are the same.
- Taxes do not need to be calculated.
- Customs documentation is not needed.

See *User Guide: Supply Chain Management* for details.

You can also use DRP to plan supply to meet demand for multiple sites within a single domain, and to distribute demand to:

- Other sites within the current domain
- Other sites in other domains within the same database
- Other sites in connected, remote databases

The execution of DRP with domains does not require any special setup. The system determines whether database switching is needed based on the domain associated with the site in Site Maintenance.

See "Associating Domains with Sites" on page 14.

If all of your domains are located in one database, the DRP process is simplified since you never have to be concerned about database connections not being available. In this case, the following functions are not needed:

- Intersite Demand Validation (12.17.12), which is run at the supply database to search for all changes to intersite demand that occurred in the demand database while the database connection was not available
- Intersite Demand Transfer (12.15.9), which is used to transfer system-generated intersite requests to the supply site's database
- Intersite Demand Export (12.15.10), which is used to place demand records in an ASCII file to send to the supply site
- Intersite Demand Import (12.17.10), which is used to import demand records in an ASCII file into the supply site's database

Using Enterprise Operations Planning

Functions in the Enterprise Operations Planning (33) module let you plan for end items and family items for multiple sites both within a single database and across multiple connected databases. With Shared Services Domain, you can also execute planning functions for sites in multiple domains within a single database.

See *User Guide: Supply Chain Management* for details.

The execution of Enterprise Operations Planning with domains does not require any special setup. The system determines whether database switching is needed based on the domain associated with the site in Site Maintenance.

See "Associating Domains with Sites" on page 14.

Using Features Across Domains in a Database

The previous section describes functions that work across databases and domains. This section describes functions that let you view and manage data across domains only. These functions include:

- Two reports that display transaction numbers across domains
- Functions for managing batch requests

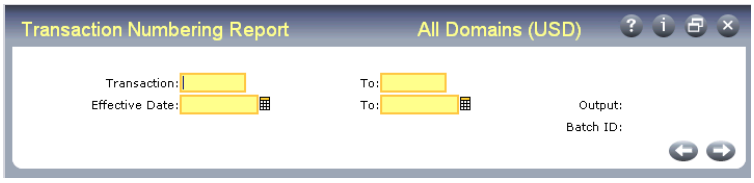
Viewing Transaction History Across Domains

If your database has multiple domains, you can use two reports to review transaction numbers in sequence. Since sequence numbers apply to the database as a whole, transactions within a domain may appear to have gaps. This report lets you see transactions created in all domains and verify that numbering is sequential.

- Use Transaction Numbering Report (3.21.19) to review inventory transaction history by number or date range.
- Use Operations Numbering Report (17.13.22, 18.4.16, and 18.22.4.12) to review operation transaction history by number or date range.

Figure 2.10 illustrates Transaction Numbering Report. The Operation Transaction Report is very similar.

Fig. 2.10
Transaction Numbering Report (3.21.19)



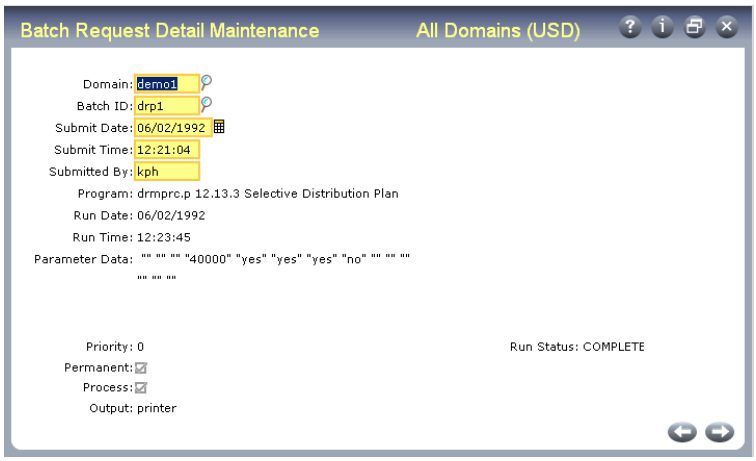
Managing Batches

Functions on the batch menu (36.14) facilitate the management of batch jobs in a database with multiple domains. System administrators can edit and process batch jobs from multiple domains without having to switch the current working domain associated with their user IDs.

Updating Batch Request Detail

Batch Request Detail Maintenance (36.14.3) has lets you specify the domain with batch requests you want to modify. Any domain you specify must be associated with your user ID in User Maintenance.

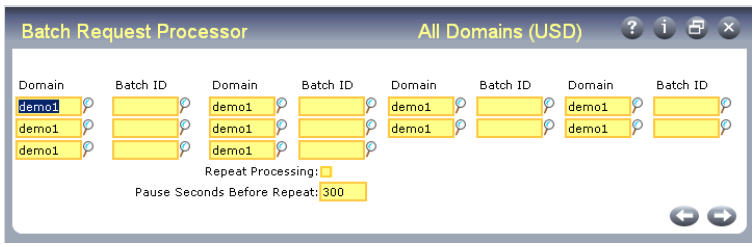
Fig. 2.11
Batch Request Detail Maintenance (36.14.3)



Submitting Batch Requests

Batch Request Processor (36.14.13) lets you specify each domain associated with the 10 batch IDs that can be processed at the same time.

Fig. 2.12
Batch Request Processor (36.14.13)



Batch Request Browse (36.14.4) lets you quickly review requests that have been submitted for a range of domains. The Batch Request Detail Report (36.14.5) also supports selecting detail by a range of domains.

Chapter 3

Domain Constants

The programs on the Domain Constants menu control calendars and codes used within a domain.

Overview

Domain constants provide basic data used throughout the system. All codes defined by the functions listed in Table 3.1 are domain specific. Since a domain represents a distinct business operation, codes can be quite different between domains. If you need to use the same code in more than one domain, you must set it up for each domain that requires it.

Table 3.1
Domain Constants Menu (36.2)

Number	Menu Label	Program
36.2.1	Holiday Maintenance	mghdmt.p
36.2.2	Holiday Browse	mgbr017.p
36.2.5	Calendar Maintenance	mgscmt.p
36.2.6	Calendar Inquiry	mgsciq.p
36.2.9	Rounding Method Maintenance	mgrndmt.p
36.2.10	Rounding Method Browse	adbr016.p
36.2.11	Rounding Method Report	mgrndrp.p
36.2.13	Generalized Codes Maintenance	mgcodemt.p
36.2.14	Generalized Codes Browse	mgbr004.p
36.2.17	Reason Codes Maintenance	mgrnmt.p
36.2.18	Reason Codes Browse	mgbr007.p
36.2.19	Reason Codes Report	mgrnrp.p
36.2.21	Number Ranges Menu ...	
36.2.21.1	Number Range Maintenance	nrsqmt.p
36.2.21.2	Sequence Browse	nrbr001.p
36.2.21.5	Sequence Number Maintenance	nrnxmt.p
36.2.21.13	Sequence Number History Report	nrsqrp.p
36.2.21.23	Sequence Delete/Archive	nrsqup.p
36.2.22	Change Tracking Maintenance	mgtblcmt.p
36.2.23	Change Tracking Browse	mgbr223.p

Maintaining Holiday and Shop Calendars

The shop calendar is required for planning, manufacturing, and distribution modules. The calendar indicates what days the plant is open and how many hours are worked each day. This information is used:

- To schedule start and due dates for MRP planned orders, master schedule orders, and work orders
- To schedule operations for work orders and repetitive schedules
- To schedule the procurement or shipment of materials through association with suppliers and customers

Use Calendar Maintenance (36.2.5) and Holiday Maintenance (36.2.1) to maintain the calendars.

Calendar Maintenance

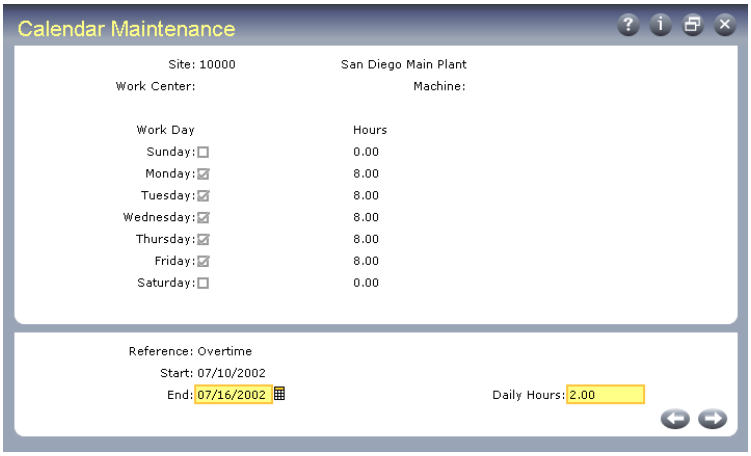
Use Calendar Maintenance (36.2.5) to specify normal work days and normal work hours for each site and its work centers. You create *shop* calendars for manufacturing using Calendar Maintenance, but you use Customer Calendar Maintenance (7.3.1) to create customer calendars. At least one calendar must exist.

You can create unique shop calendars by specifying some fields while leaving others blank. A default shop calendar has a blank site, work center, and machine. The system searches for a shop calendar in the following order:

- For the specific site, work center, and machine combination
- For site and work center with a blank machine
- For site with both work center and machine blank

If shift patterns vary because of overtime, increased or reduced shifts, or plant shutdowns, enter exception hours. Set up exceptions for a date range by specifying the number of hours that are added to or subtracted from normal work hours.

Fig. 3.1
Calendar Maintenance (36.2.5)



In a calendar, work days are marked with a Yes and nonwork days with a No. Manufacturing order due dates are scheduled only on work days. Each work day has a production capacity in hours. This should exclude breaks and nonproductive time. Manufacturing operations can be scheduled only up to the production capacity of the day.

Shop calendars are typically defined in this order:

1. Create a system calendar by leaving site and work center blank.
2. Create a calendar for each site with blank work centers. CRP uses this calendar to calculate capacity, including holidays.
3. Create work center calendars with site and work center filled in.

The system searches for a calendar from the most specific to the least specific—specific site, work center, and machine combination first and blank site, work center, and machine last.

You can specify exceptions, such as overtime or machine downtime for preventive maintenance. The system uses exception information only when preparing operation schedules, but not when calculating manufacturing order due dates.

Example On July 14, two hours of overtime are scheduled at site 10000. Enter OVERTIME as the reference code, July 14 as both start and end dates, and +2 as Daily Hours.

If an exception occurs on a day that is not part of the standard work week, add that exception to an existing day rather than changing the standard work week. Many scheduling programs assume that the work week has a certain number of days. Adding a day to the standard work week can result in inaccurate schedules.

Holiday Maintenance

Use Holiday Maintenance (36.2.1) to schedule holidays and other nonwork days that apply to the entire site.

Fig. 3.2
Holiday Maintenance (36.2.1)

Holidays are days that no one works; the plant is shut down and no production is scheduled. Manufacturing orders are never due and operations are not scheduled on a holiday.

Defining Rounding Methods

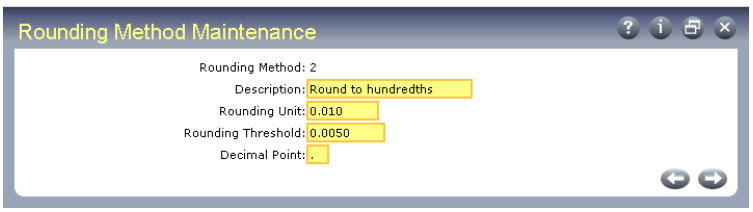
The system lets you round monetary amounts in a manner consistent with a given currency.

Three rounding methods exist by default:

- 0. Round to zero decimals, using 0.5 as the rounding threshold.
- 1. Round to one decimal, using 0.05 as the rounding threshold.
- 2. Round to two decimals, using 0.005 as the rounding threshold.

You can set up additional rounding methods as needed in Rounding Method Maintenance (36.2.9). After defining rounding methods, apply them to currencies in Currency Maintenance (26.1).

Fig. 3.3
Rounding Method Maintenance (36.2.9)



Rounding Method. Enter an alphanumeric code identifying the new rounding method to be created.

Rounding Unit. Enter the number of decimal places to which monetary values are rounded. For example, to specify rounding to three decimal places, enter 0.001.

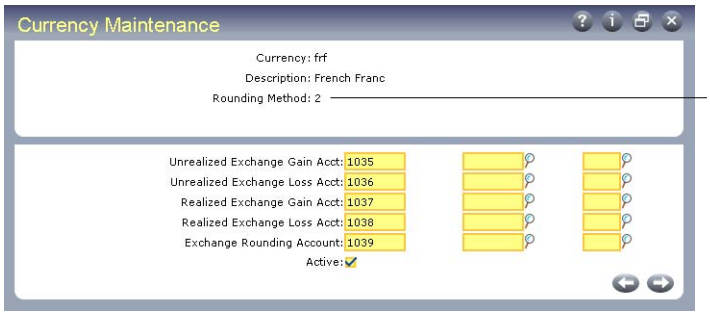
Rounding Threshold. Enter the number at which monetary values are rounded up. This number must be less than the number entered for the rounding unit.

For example, if the rounding unit is 0.001, entering 0.0025 for the rounding threshold tells the system that decimal values of 25 ten-thousandths and higher are to be rounded up to the nearest one-thousandth. Amounts are rounded based on their absolute value. For example, –9.99 is rounded the same as 9.99.

Decimal Point. Enter the character to be used as the decimal point in monetary values.

Use Currency Maintenance (26.1) to apply rounding methods to currencies.

Fig. 3.4
Currency Maintenance (26.1)



Review the rounding methods you define using Rounding Method Browse (36.2.10) or Rounding Method Report (36.2.11).

Establishing Generalized Codes

When you install a new database, a number of system and reference fields accept any kind of data, as long as it does not exceed the field length. You can customize the user interface by adding generalized codes and lookups.

Before implementing a module or a particular functional area, the implementation team should determine which fields should have generalized codes and lookups.

Generalized codes are domain specific since these codes may vary widely based on the type and location of the business operation. For example, customer types, sales distribution channels, and buyer/planner codes could differ between a domain representing a business in England and one in Germany.

Important Some programs that update system-wide data such as User Maintenance (36.3.1) reference generalized codes. These generalized codes must exist in all domains or you may encounter errors editing a user record depending on what your current working domain is.

When using generalized codes, you can control three different conditions:

- What the acceptable values in a field are. Define these values in Generalized Codes Maintenance (36.2.13).
- Whether a list of acceptable values displays in a look-up browse on the field. Specify this in Drill Down/Lookup Maintenance (36.20.1).

- Whether the codes you have created are the only acceptable codes (that is, whether the list is validated). This may require you to add a validation expression to the data dictionary. See “Adding Validation” on page 29.

Field Validation

Before entering a list of generalized codes for a field, you must know the field’s name and size. In the Windows and character interfaces, a pop-up window displays information about the field when you press Ctrl+F with your cursor in the field. If this information indicates generalized codes validation, the system automatically verifies data entered in the field against the list of generalized codes.

You can also use Generalized Codes Validation Report (36.2.15) to view a list of all fields in the database that have schema validation assigned. This is the preferred method in the QAD .Net UI and Desktop interfaces.

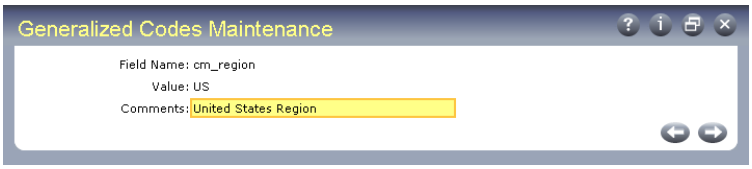
Note The system performs validation only when generalized codes have been defined for a field.

Example You have divided your customers into regions. The cm_region in the customer master is updated by Customer Maintenance (2.1.1). As part of the implementation process, you assign each customer to one of two regions. To ensure that only standard region codes are used, define them as generalized codes. Specify cm_region for the field name, the values US and X-US for the two regions.

Adding Generalized Codes

Figure 3.5 illustrates Generalized Codes Maintenance (36.2.13).

Fig. 3.5
Generalized Codes Maintenance (36.2.13)



Specify a field name and then enter valid values and comments. Values cannot exceed the length of the field. The comment displays next to the value in the lookup.

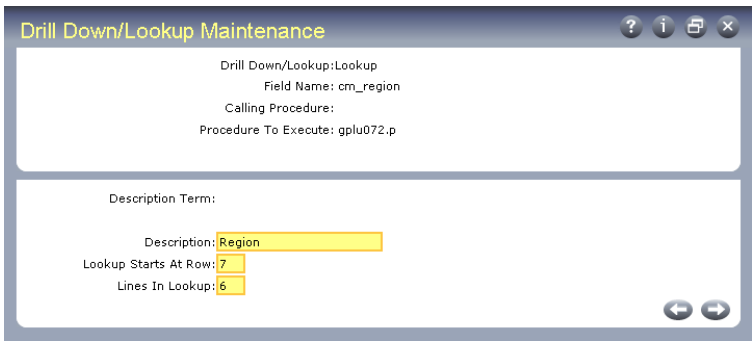
Adding a Lookup

To set up a lookup to display generalized codes, use Drill Down/Lookup Maintenance (36.20.1). Enter the field name where you want the lookup and `gplu072.p` as the procedure to execute.

See “Maintaining Drill Downs and Lookups” on page 118.

This program creates the lookup with values from the assigned field. If the lookup should only be accessed from a particular screen, enter that program name as the calling procedure.

Fig. 3.6
Drill Down/Lookup Maintenance (36.20.1)



The description defaults from the data dictionary, but can be changed here. If no description exists, the field name is a local variable. The description displays as the title of the lookup.

Adding Validation

Generalized code validation, like field security, requires a special validation expression in the database dictionary that references the file `gpcode.v`.

Some fields already reference `gpcode.v`. These display in the Generalized Codes Validation Report. If you want to activate generalized code validation for other fields, you must change the data dictionary.

You can do this directly using full Progress or, if you have encrypted source, you can use the utility `utdbfx70.p`. Once you have added a validation expression, you must recompile the affected programs. For instructions on how to do this, refer to the *Progress Programming Handbook*.

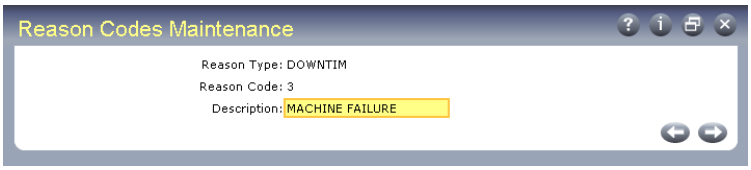
To add validation for a local variable, you must insert the validation directly in the source code.

Important If you change the data dictionary, keep careful records and be prepared to repeat the change when new product versions that update the data dictionary are installed.

Using Reason Codes

Reason codes are used in security functions, sales quotes, sales order maintenance, purchase order returns, shop floor reporting, repetitive reporting, and the Product Change Control (PCC) module. They are also used if you have enabled change tracking and in several optional modules, such as WIP Lot Trace, Electronic Signatures, and Shipment Performance. Add other custom uses as needed.

Fig. 3.7
Reason Codes Maintenance (36.2.17)



- Use codes of type User_Act for the Active Reason field in User Maintenance (36.3.1) and the Auto-Deactivation Reason field in Security Control (36.3.24).

- Use codes of type ESIG to indicate why a user is authorizing the data in an e-signature enabled program.
- Use codes of type QUOTE in the Reason Lost field of sales quotations.
- Use codes of type DOWN or DOWNTIME in the Reason field of labor feedback programs (17.1–17.4).
- Use codes of type ORD_CHG to associate changes made in Sales Order Maintenance to order detail, such as a change to the order line quantity or due date. See “Tracking Changes” on page 38.
- Use codes of type DOWN, DOWNTIME, REJECT, REWORK, ADJUST, and SCRAP for reporting in Repetitive and Advanced Repetitive programs. Use these same codes with the optional
manufacturing WIP Lot Trace module.
- Codes used in the PCC module are user-defined. They specify severity levels related to approval of change documents.
- Use codes of type SHIPQTY and SHIPTIME with the PRO/PLUS Shipment Performance module. See *User Guide: PRO/PLUS*.
- Use codes of type RTV (return to vendor) to define reasons entered in Purchase Order Returns (5.13.7).

Generate reports on downtime organized by reason code using the Downtime by Reason Report (17.17).

Managing Number Ranges

Some countries impose sequencing requirements related to tax filings or statutory reporting. In many countries, companies are legally required to prevent gaps in the numbering of official documents.

Additionally, certain business practices require different business units within the same corporation to maintain separate sequencing for similar documents such as invoices, purchase orders, sales orders, and vouchers.

Example In Italy, the number of an official document is strictly related to the date the document was printed, and it is a common practice to have multiple number ranges for shipment and invoice documents. In Brazil, the number of an official document is related to a specific physical site, requiring multiple number ranges with a prefix identifying a site code.

Number range management (NRM) supports varied sequencing requirements on a global scale. Features include gap control and multiple number ranges for the same document type.

NRM Overview

NRM generates sequence numbers built from one or more segments, each with its own set of characteristics and behavior.

You can add or remove segments during sequence definition, but once a sequence has been used to generate or validate numbers, you cannot change its structure.

Figure 3.8 illustrates a sample sequence with five segments: three fixed-value segments (NY and two dashes), one incrementing integer segment (1234), and one date-driven segment (06:15:07).

Fig. 3.8
Example Sequence Number

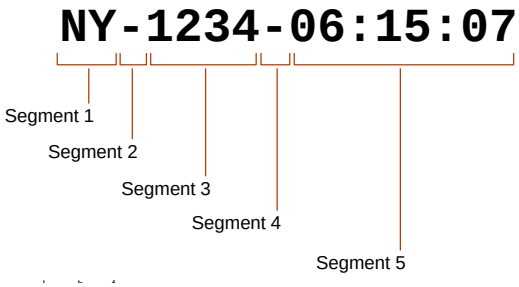


Table 3.2 describes the three segment types.

Table 3.2
Segment Types

Segment Type	Description	Required
Incrementing Integer	A range of values, with a lower bound, an upper bound, initial, and reset value.	Yes. Each sequence number must have one and only one incrementing integer segment.
Date-Driven	A value that depends on the transaction effective date or the fiscal period that corresponds to the effective date. The format is a compound string that allows the optional display of date components such as year, month, week, day, including delimiters between components. Delimiters separate the individual components of a segment. For example, 06:15:07 uses colons as delimiters.	No. Each sequence can have one date-driven segment.
Fixed-Value	Any printable character except a comma. For example, NY may be a fixed-value segment assigned by a client in New York. A fixed-value segment is not changed in any way during sequence number generation.	No.

Sequence Number Generation

To update a sequence number, the system examines each segment separately. Only date-driven or incrementing integer segment types are modified. A fixed-value segment is never changed.

Control Segments

You can set up a date-driven segment as a control segment. In this case, changing its value causes the incrementing integer segment to reset to its assigned reset value. When a control segment does not exist or does not change, the incrementing integer segment is incremented.

Sequence Parameters

Create sequence numbers and define sequence parameters using Number Range Maintenance (36.2.21.1). A distinct segment editor defines the format and parameters of each segment type.

Internal and External Sequences

There are two types of sequence number: internal and external.

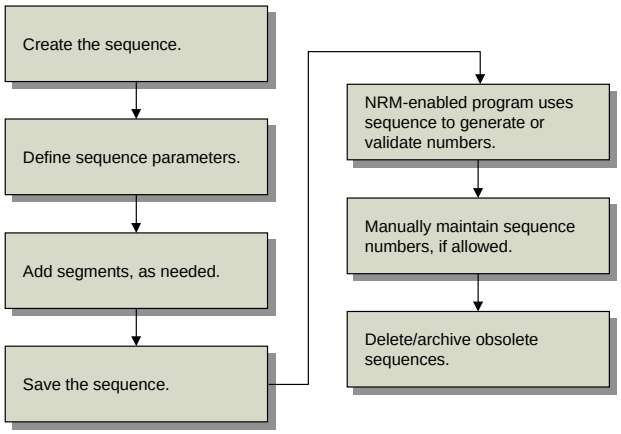
Internal sequences automatically generate numbers in ascending order as needed. NRM examines each segment in the sequence to determine whether to update its value. A fixed-value segment remains unchanged during sequence number generation.

External sequences accept a sequence number entered externally and validate it against a sequence definition. NRM verifies that the number belongs to the set defined by the sequence and that it has not yet been used. The system parses the number into segments and validates each segment against the corresponding segment in the sequence definition.

Sequence Life Cycle

Figure 3.9 illustrates the life cycle of a sequence.

Fig. 3.9
Sequence Life Cycle



To set up a sequence, create an ID, define general parameters, and add appropriate segments. Once a sequence is defined, a program uses it either to obtain a new number or validate user-entered numbers.

Note Programs must be specially designed to use NRM sequence numbers.

If you attempt to discard or void a number, the system checks the sequence definition to ensure that this is allowed.

You can delete and archive unneeded sequences.

NRM Sequences

Programs must be specifically enabled to use NRM. Currently, NRM sequences are used in general ledger (GL) daybooks, fixed assets, logistics accounting, shipping, the PRO/PLUS WIP Lot Trace module, and Kanban.

Fixed Assets

An optional NRM sequence number can be specified in Fixed Asset Control (32.24) for automatically generating fixed asset ID numbers.

See *User Guide: Fixed Assets*.

General Ledger Daybooks

GL daybooks let you group and report GL transactions. Unposted transactions include the daybook code and daybook entry number. NRM generates entry numbers based on the ID of the daybook.

See *User Guide: Financials*.

Logistics Accounting

If you are using the optional Logistics Accounting module, two NRM sequences must be defined in Logistics Accounting Control (2.15.24) for distribution order shipments and sales order shipments.

See *User Guide: Master Data*.

Shipping

Many countries legally require businesses to maintain strict control when assigning numbers to shipping documents. This is also true when multiple number ranges are assigned to the same type of shipping document. To meet this need, NRM is required for all shipper functionality.

See *User Guide: Distribution*.

WIP Lot Trace

An optional NRM sequence number can be specified in WIP Lot Trace Control (3.22.13.24) for generating WIP lot and serial numbers in the various functions that trace them.

See *User Guide: PRO/PLUS*.

Kanban

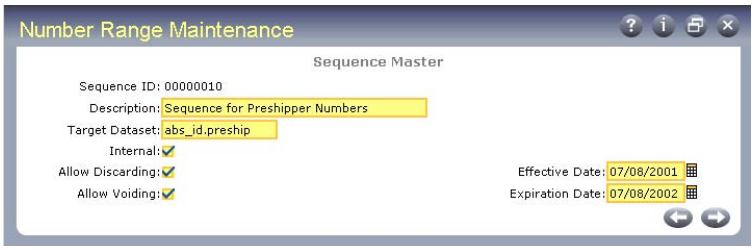
If you use dispatch lists to communicate kanban card authorizations to your suppliers, you must specify an NRM sequence in Kanban Control (17.22.24). The system uses the sequence to generate an ID number during dispatch list processing.

See *User Guide: Kanban*.

Setting Up Sequences

Create sequences and define sequence parameters using Number Range Maintenance (36.2.21.1). NRM uses a unique sequence ID to retrieve data and generate new numbers. Use Sequence Browse (36.2.21.2) to view the defined structure of a sequence.

Fig. 3.10
Number Range Maintenance (36.2.21.1)



Sequence ID. Enter a code uniquely identifying a sequence. Create a new sequence or use Next/Previous to retrieve an existing sequence.

Description. Enter a description of this sequence, up to 40 characters.

Target Dataset. Enter the dataset identifier associated with this sequence. The target dataset can indicate who owns the sequence or where its numbers are used. A sequence owner can be a process, a document, or any other entity that the client program can recognize.

Note The target dataset could be the name of the principal database field where numbers from the sequence are used.

You cannot create a new sequence that intersects an existing sequence with the same target dataset—creating two sequences that could generate the same sequence number for the same target field.

For example, if sequences A and B both target field so_nbr, they cannot have a common element that could cause conflicts.

The following three target datasets are used with shippers:

- abs_id.shipper is used for sales order shippers.
- abs_id.preship is used by sales order pre-shippers.
- abs_id.mbol is used by master bills of lading.

For Fixed Assets, specify dataset fa_id.

For Logistics Accounting, specify:

- la_so_ship_id for sales order shipments
- la_do_ship_id for distribution order shipments

For Kanban, specify dataset knbd.dispatch_id.

Internal. Specify whether the sequence numbers are supplied by an external source or automatically generated by NRM. Enter Yes if numbers are generated by NRM.

Allow Discarding. Using a number causes it to be registered. This field determines whether a registered number can be discarded, leaving a gap in the sequence.

No (the default): Gaps are not allowed and numbers cannot be discarded from this sequence.

Yes: You can discard previously registered numbers from this sequence by reversing the register operation. NRM then erases all record of the sequence number, and the discarded number is replaced by a gap.

Allow Voiding. Determines whether you can mark a registered number as void.

No (the default): Numbers in this sequence cannot be voided.

Yes: You can void numbers and specify a brief description why. Voiding is recorded as a separate event in the sequence history.

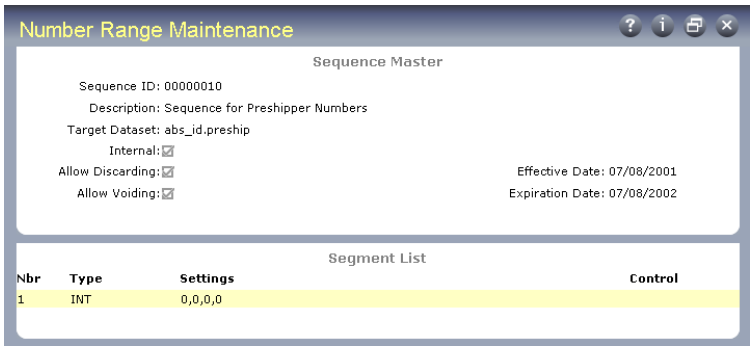
Effective Date. Indicates the earliest date when this sequence can be used.

Expiration Date. Indicates the latest date when this sequence can be used.

Segment List

After you define the initial parameters for a sequence, Segment List and Editor frames display. The segment list shows the type and settings for each segment defined in the sequence. Segments display in ascending order, based on segment number.

Fig. 3.11
Number Range Maintenance (36.2.21.1), Segment List Frame



Segment Editors

The segment editor used depends on the type of segment being defined. Use the editor to create or modify the segment format definition and assign a new segment number. There are four types of segment editors.

- *Fixed* segment editor for fixed value segments
- *Integer* segment editor for incrementing integer segments
- *Date* segment editor for date-driven segments
- *Fiscal* segment editor for date-driven segments, relative to fiscal periods

Fixed Segment Editor

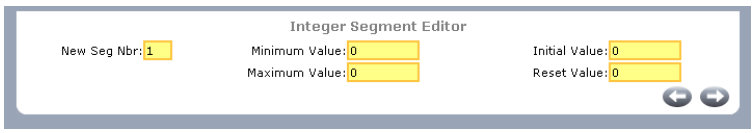
Use the fixed segment editor to establish a fixed string value. You can use any printable character except a comma.

Fig. 3.12
Fixed Segment Editor

Integer Segment Editor

Use the integer segment editor to specify the initial, reset, minimum, and maximum values for a segment.

Fig. 3.13
Integer Segment Editor



Date Segment Editor

Use the date segment editor to tell NRM how to display a date component of a sequence number. Specify codes representing date components such as year, month, day. You can add components in any order, with optional delimiters. In the date segment 07/02, a forward slash is the delimiter. You can use any printable character except a comma or another date component as a delimiter.

You can indicate if this segment is a control segment. Changing the value of a control segment causes the incrementing integer segment to reset to its assigned reset value. The new value in the control segment ensures that the sequence numbers generated after resetting are unique within the target dataset.

Fig. 3.14
Date Segment Editor

Fiscal Segment Editor

Use the fiscal segment editor to tell NRM how to display a fiscal date component of a sequence number. Codes represent a component of a fiscal period. Otherwise, this editor is exactly the same as the date segment editor.

Note You can add fiscal segments only if the sequence has an expiration date.

Fig. 3.15
Fiscal Segment Editor

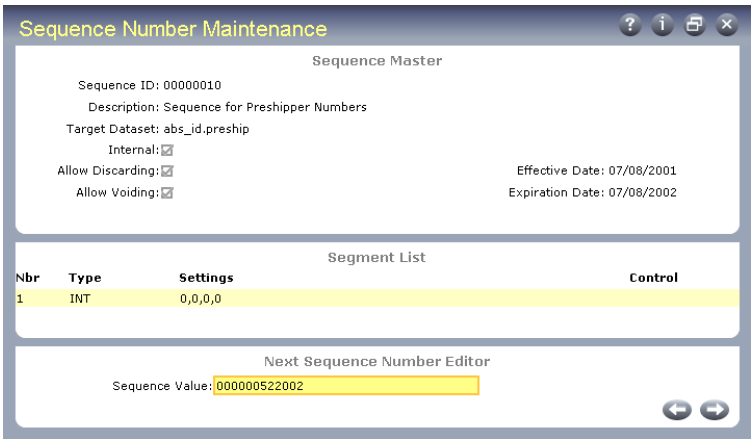
Setting Sequence Values

Use Sequence Number Maintenance (36.2.21.5) to set the next value for an existing sequence, when:

- The sequence is internal.
- Allow Discarding is Yes.

The default in Sequence Value is the last number that was used. If you update it, the system validates the new value against the segment order and settings. It then increments the new value the next time the sequence is used.

Fig. 3.16
Sequence Number Maintenance (36.2.21.5)



Viewing Sequence Number History

When a client program uses a sequence to dispense or validate numbers, the system creates history records. Use Sequence Number History Report (36.2.21.13) to view history data on internal and external sequences.

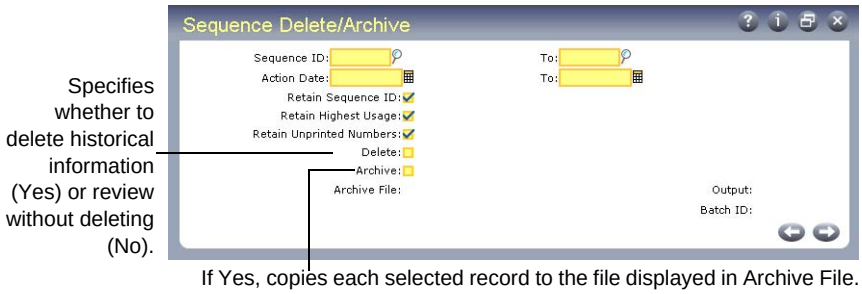
You can view the sequence definition, which sequence numbers have been used, and which sequence numbers have not been used, including gaps. This report helps you to identify missing documents by reporting numbers that are not recorded in the sequence history.

Deleting and Archiving Sequences

Use Sequence Delete/Archive (36.2.21.23) to delete sequences and associated history. You can optionally archive information to an external file and later restore it using Archive File Reload (36.16.5).

Once sequence history is deleted, it no longer appears on the Sequence History Report.

Fig. 3.17
Sequence Delete/Archive (36.2.21.23)



Tracking Changes

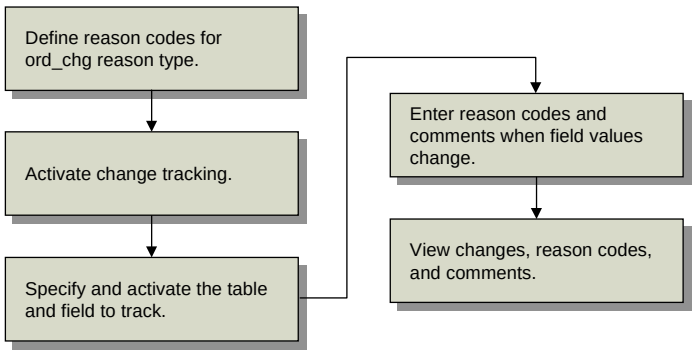
Use Change Tracking Maintenance (36.2.22) to mark sales order detail fields for change tracking. For line detail information in discrete sales orders, you can:

- Specify which field to track.
- Activate or deactivate tracking.
- Delete any records for fields that no longer require tracking.
- Allow users to enter a reason code and comments when the value of a marked field changes.
- Print the changes, reason codes that explain the changes, and any associated comments on a Booking Transaction Report (7.15.14). See *User Guide: Distribution.*

Change Tracking Implementation Overview

When implementing change tracking, you work with different programs to set up codes, activate change tracking, specify what to track, then view results. Figure 3.18 illustrates the basic change tracking implementation flow.

Fig. 3.18
Change Tracking Implementation Flow



Defining Change Tracking Reason Codes

You must define reason codes that explain changes to sales order detail in Reason Code Maintenance (36.2.17). You specify ord_chg as the reason type. You must define at least one reason code for the ord_chg type to implement change tracking.

Create reason codes that fit your company's most common reasons for changes to sales order details. For example, you can create a Delete reason code for deleted orders.

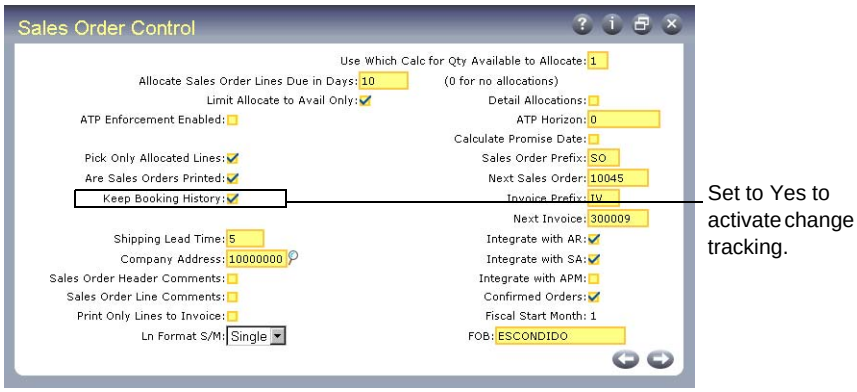
See "Using Reason Codes" on page 29.

Activating Change Tracking

Activate change tracking by setting Keep Booking History to Yes in Sales Order Control (7.1.24).

See *User Guide: Distribution*.

Fig. 3.19
Sales Order Control (7.1.24)



Specifying Fields to Track

Use Change Tracking Maintenance (36.2.22) to:

- Specify which table contains the fields you want to track.
- Specify which fields to track.
- Delete any records for fields that no longer require tracking.

Fig. 3.20
Change Tracking Maintenance (36.2.22)

Table. Enter the database table that contains the field that is being tracked for changes. Currently, Change Tracking Maintenance tracks only the sales order detail (sod_det) table.

Description. Enter a brief description (24 characters) of the database table.

Active. Specify Yes to track changes for the database table you specified. Specify No to deactivate tracking. The default is No.

You must set Active to Yes for both the table and the field before change tracking begins.

Delete. Specify Yes to display the reason code pop-up in Sales Order Maintenance when the user deletes an entire sales order line. Specify No if you do not want the reason code pop-up to display. The default is No.

Note You must set Active to Yes and specify a field to track.

Once you complete these fields and press Go, the following frame appears.

Fig. 3.21
Change Tracking Maintenance, Field Frame

Field. Enter the field to track. Currently, you can only track fields belonging to the sales order detail (sod_det) table.

Note To find the field name in the character or Windows user interface, press Ctrl+F while your cursor is located in the field. In QAD .Net or Desktop, the field name displays as a field tip when your cursor moves over a field.

Description. Enter a brief description (24 characters) of the field.

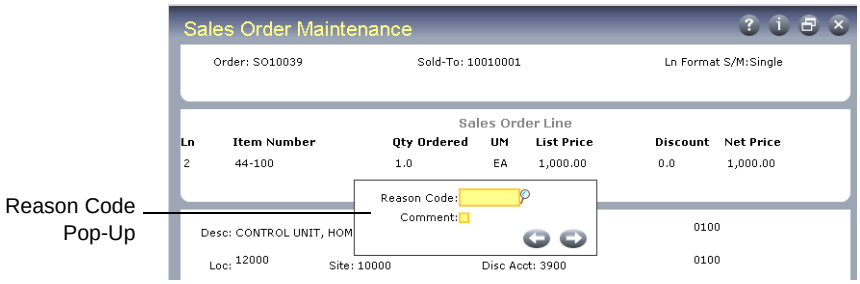
Active. Specify Yes to activate tracking for the field you specified. Specify No to deactivate tracking. The default is No.

Review the tables and fields you specify and their active or delete status using Change Tracking Browse (36.2.23).

Reason Code Pop-Up

After you activate change tracking and specify a table and field to track, when the user changes or deletes the value of the field, a reason code pop-up displays. Currently, only the sales order detail table can be tracked; therefore, the reason code pop-up displays in Sales Order Maintenance (7.1.1).

Fig. 3.22
Reason Code Pop-Up in Sales Order Maintenance (7.1.1)



Select a code that indicates the reason you are changing the value of the field or deleting the line. The reason type associated with the code must be ord_chg.

Even though you can track multiple fields, you are only prompted once with the reason code pop-up. Use the comment screen to explain multiple changes you made to the sales order line.

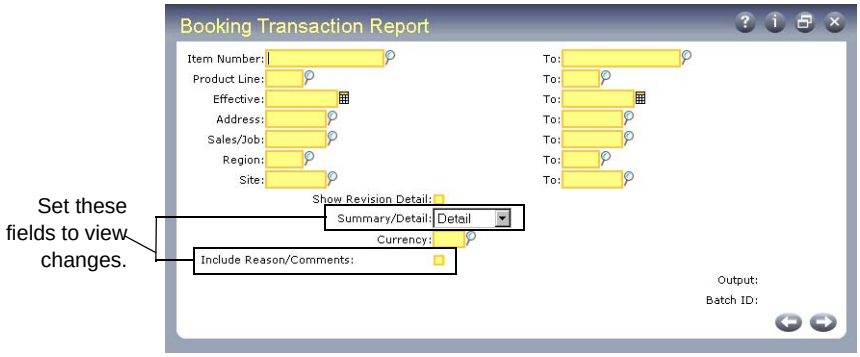
Viewing Changes

To view changes you tracked, use Booking Transaction Report (7.15.14). The report displays the reason and comments related to a discrete sales order line change.

To display the changes, set:

- Summary/Detail to Detail
- Include Reason/Comments to Yes

Fig. 3.23
Booking Transaction Report (7.15.14)



Chapter 4

System Interface

The System Interface menu contains programs that control menus, messages, multi-language installations, and help.

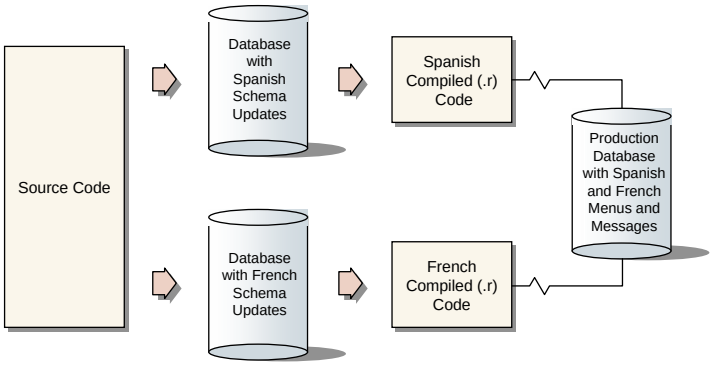
Note If you are using the QAD .NET or Desktop user interface, interface details are discussed in the associated user guide.

Using Multiple Languages

The system supports multi-language capabilities in two areas:

- Screens displayed in multiple languages
- Data stored and displayed in multiple languages

Fig. 4.1
Multiple Language Installations



The system can display screens in multiple languages because the programs are in multiple languages. If you have some users who want to see screens in Spanish and others who want to see them in French, you need a set of Progress programs in each language. The Spanish programs are compiled against an empty database with Spanish data definitions (labels and validation messages). The French programs are compiled against a second database with French data definitions.

The system can display menus, messages, and field help in multiple languages. The standard menus and messages are in the production database. Field help is in the field help database, `mfghelp.db`.

The fact that the Progress programs are in multiple languages does not affect the production database. To retrieve data in multiple languages, each piece of information in the production database must be stored once in each language.

Most orders include comments, which often must be in multiple languages. These can be stored in multiple languages and retrieved by language ID. You can also customize menus and messages and assign a language ID so the system knows which entry to display.

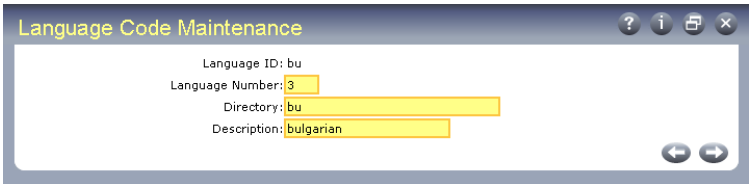
However, not all data in the system can be stored and displayed by language ID. For example, item descriptions can be stored in only one language.

Setting up Multiple Languages

To work in full multi-language mode, you must:

1 Specify the top-level directory for each language's object code in Language Code Maintenance (36.4.1).

Fig. 4.2
Language Code Maintenance (36.4.1)



This ensures that the system can locate the programs for each language. The programs for each language must be stored in separate subdirectories.

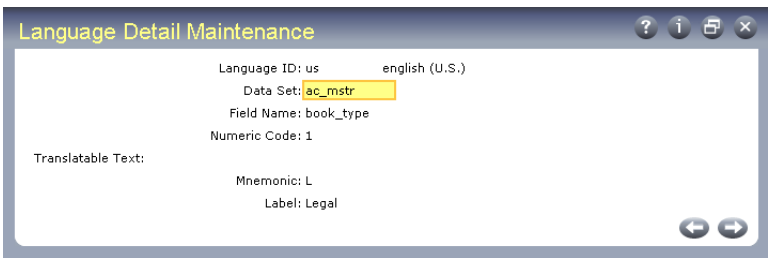
2 Designate the default language and country code for each user in User Maintenance (36.3.1). This ensures that when the user logs on, the system calls the Progress programs for that person's language. See "Defining Users" on page 155.

If the language is the same for all users but multiple language comments are required for orders, you only need to define the separate language codes in Language Code Maintenance. A number of codes for supported languages are already defined.

Language Detail Maintenance

Some program options appear on the screen using alphabetic codes or words. Internally, these options are controlled by numeric codes. Mnemonics and labels provided in English may not be appropriate in other languages. Use Language Detail Maintenance (36.4.3) to change, add, and delete mnemonic codes and labels.

Fig. 4.3
Language Detail Maintenance (36.4.3)



Data Set. Enter the program name, a database table name, or an abbreviation of the functionality for a field.

Field. Enter the field name associated with the data set.

Numeric Codes. These are the values used by the programs. A mnemonic code can be assigned for each numeric code. Codes cannot be added or edited.

Mnemonic. Mnemonic codes are already assigned for each field with several system-specified options. These codes can be changed, added, or deleted using this program.

Label. Default labels already exist for the different mnemonic codes. These labels can be changed, added, or deleted using this program.

Customizing Menus and Function Keys

You can execute a program in a number of different ways.

- Type the program name, such as `mgmemt.p`, at any menu prompt. When you exit the program the prompt redisplays.
- Type the full number, such as 36.4.4, at any prompt. If you are currently on another branch of the menu tree (for example the 1.4 menu), enter a period before the menu number (.36.4.4).
- Type a partial number from a submenu, such as 4.4 while located at menu 36.
- Press a function key that is assigned to this program.
- Select the program from the User Menu.

You can control the menu numbers and the names associated with programs in several ways.

- Move menu items.
- Change menu names.
- Create names for menu items.
- Specify security for menus. See “Assign Access by Menu” on page 168 security chapter.

Note If you make these changes, they may be lost during software updates.

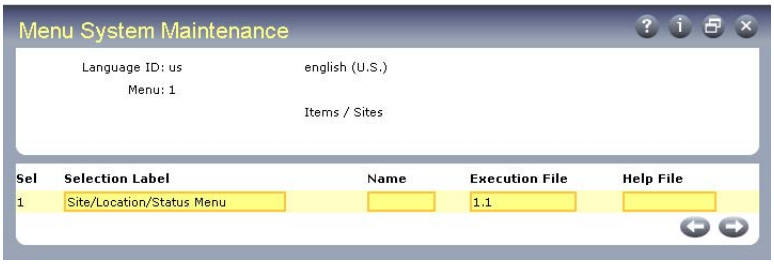
All menu information is contained in the mnd_det table. View its structure in the Data Dictionary. With each release, you receive the latest version of this table, which you should load into your databases. As QAD develops new programs, it populates this table with new records and alters existing records. When loading the latest version of the table data, you must delete your existing version—along with any modifications you made.

The new menus are loaded with a utility program `mgdload.p`, which provides some control over what gets replaced and prints a listing of what was changed. The mnd_det table is modified by two programs: Menu System Maintenance (36.4.4) and User Function Maintenance (36.4.11).

Menu System

Use Menu System Maintenance (36.4.4) to assign menu labels and execution files to menu numbers. When users type the number, the file executes. If you want to move a menu item or have it execute a different program, change the record with this program.

Fig. 4.4
Menu System Maintenance (36.4.4)



The Name field allows you to call programs using keywords. For example, for a program buried deep in the menu structure, you can add a name and then execute the program by typing that name on any menu command line.

Note If you are using QAD Desktop, you must use Desktop utilities to rebuild the menus and the search database whenever you add new menu items or change existing ones. Otherwise, your changes will not be visible to users. See *Installation Guide: QAD Desktop*.

User Menu and Function Keys

Assigning function keys to frequently used menu items is another way to execute programs quickly. Keys can be established for all users or individually customized. Up to 999 function keys can be defined. In addition, you can change the standard label for a menu item to customize menu labels for each user.

The effect of the records you define in User Function Maintenance (36.4.11) varies depending on the selected user interface. Function keys apply only to the character interface. However, the programs assigned to function keys also display on the User Menu in the character and Windows interfaces and under the My Programs link in QAD Desktop.

Example Use of Function Keys

A user entering a sales order may need to check on the available-to-promise (ATP) quantities for an item before indicating a due date. By setting up a function key for the Master Schedule Summary Inquiry (22.18), the order clerk can review an item's ATP quantity without leaving Sales Order Maintenance (7.1.1).

Note Do not use function keys or the function menu to access a maintenance screen in the character or Windows environments. Progress only completes transactions initiated with function keys after the initial transaction is completed. If, for example, you are in sales orders, you start an order, then perform an inventory transaction using a function key, and then cancel the sales order, the inventory transaction is also canceled.

Windows Interface

Access user functions from the pull-down User Menu. This menu has multiple sections:

- User menu items display in the top section, ordered by the value of the Function Key and Sequence fields. For example, the program assigned to function 13, sequence 2 follows the program assigned function 13, sequence 0. The program assigned to function 15 comes after both of these.
- Programs defined in User Tool Maintenance (36.20.4) display below User Menu items. They also display as buttons on the toolbar of programs with which they are associated. Unlike user menu items, you can associate user toolbar items with specific programs or groups of programs. See "User Tool Maintenance" on page 122.

Note Programs defined with User Tool Maintenance do not display on browses.

The exact menu items that display depend on whether you have user-specific items defined in User Function Maintenance.

- If you have user-specific items defined, they display on the menu.

- If no items are associated with your user ID, the menu includes only items assigned to a blank user ID.

Note This is unlike the character interface, where users can see both menus.

Character Interface

Access programs associated with a function key by selecting that function key. Function keys F1 through F12 are reserved for system use, so the assigned key must be F13 or higher. Since many keyboards do not handle that number of function keys, this option is used less frequently.

User Menu in Character Interface

Access the User Menu by pressing F6. A list of menu items set up for your user ID appears. Choose the one you want by highlighting it and pressing Enter or Go. Press Tab to sort the list by menu number or function name. Press End to display the user menu items defined without a user ID.

Note There is no relationship between the order of items on the User Menu and the function key assigned, and the function key is not shown. Menus sort lexically, so that 13 appears before 2 if you are in the Menu Selection column.

Different environments have different function key uses and limitations. Set up your system according to your environment. For example, if your system is limited to only 12 function keys, do not attempt to use the function keys as a quick method to launch programs. Instead, use the User Menu.

Executing Programs in Sequence

In the character interface, you can make several programs execute in sequence by assigning them to the same function key and giving each a different sequence number. When you press that function key, the first function in the sequence executes. When that function is finished, the next one in sequence is called automatically.

Important All transactions in the sequence must be completed before data is updated in the database.

QAD Desktop

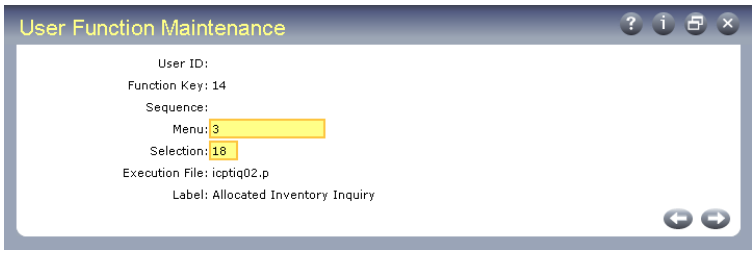
If you are using QAD Desktop, the programs you specify with User Function Maintenance display on the My Programs menu under My Desktop. In Desktop, My Programs lets you organize frequently used programs rather than being a way to access multiple programs. This is because you can always run multiple programs simultaneously in detached windows. You do not need to be concerned about running two maintenance programs at the same time.

User Function Maintenance

Set up user menus and function keys in User Function Maintenance. Each selection on the user menu should have a different function key reference, from 13 to 40, and a zero or blank sequence number. The function key reference must be 13 or greater, even if your keyboard supports fewer function keys or you plan to access selections through the User Menu.

Note To set up function keys, terminals must be compatible with the Progress protermcap file.

Fig. 4.5
User Function Maintenance (36.4.11)

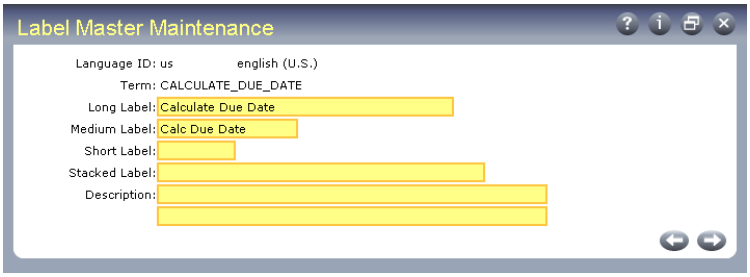


Modifying Labels

The system dynamically reads the label master table to determine the appropriate labels to display on screens and reports. For the system to display labels from the label master, Translate Frames must be Yes in Label Control (36.4.17.24). Otherwise, screens and reports display field labels statically from the source code.

You can modify how labels display in Label Master Maintenance (36.4.17.1). You may want to modify labels in order to meet specific company needs or to improve definitions of non-English labels.

Fig. 4.6
Label Master Maintenance (36.4.17.1)



The system validates the language code and accesses the *term*. The term is the key that links labels to fields, allowing the system to determine which labels to display. The term remains the same regardless of the language selected.

Terms display in all uppercase with underscores; for example, CALCULATE_DUE_DATE is the term for Calculate Due Date when the language code is US (American English).

Use Label Detail Maintenance (36.4.17.5) to assign terms and labels defined in Label Master Maintenance to fields generically or to fields in specified programs.

Warning Because terms can be assigned to fields accessed by many programs, label modifications and new term assignments should be made with extreme caution.

Modifying Messages

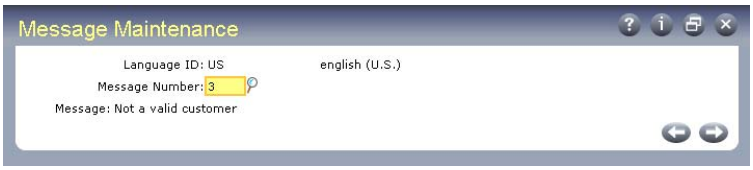
The system has two kinds of messages:

- Validation messages stored in the data dictionary. These display when the contents of the field do not match its specifications.
- Program messages stored in the database. These display in all other cases.

Numbered Progress error messages sometimes display when a Progress instruction fails. Most of these messages are handled by the system, and a program error message is substituted, so this should occur rarely.

You can modify messages in Message Maintenance (36.4.7). One reason for changing messages is multiple language requirements. If a message seems unclear to some end users, an administrator can clarify its meaning.

Fig. 4.7
Message Maintenance (36.4.7)



Changing messages can create the same version control problems that occur when menus are changed. Be careful to use message numbers not likely to be used in a later version.

Using Field and Procedure Help

The system provides two types of online help: procedure and field help. Procedure help explains what the current function or program you are working within does. Field help describes particular fields.

You can view these help records in either Windows (F1 key) or character (F2 key) format. The content of the Windows and character help files is identical. However, you can add your own information to the character help files.

In the Windows or character interface, view field help by pressing the appropriate function key with the cursor in the field. Press the key again and procedure help displays.

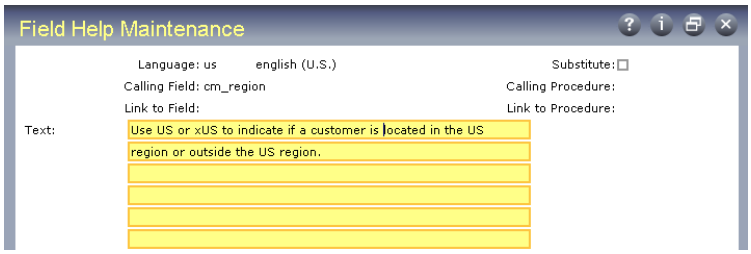
Note QAD .NET displays the character help data in an HTML format. Any changes you make to character help are also visible in those user interfaces.

For more information on help functions in the QAD .NET UI, see *User Guide: Introduction to QAD Enterprise Applications*.

Adding User Help

Use Field Help Maintenance (36.4.13) to add to the character-format help delivered with the system.

Fig. 4.8
Field Help Maintenance (36.4.13)



Custom text entered in Field Help Maintenance appears first when you press the Help key. Press Help again to display standard QAD help.

Printing Help

You can print out portions of the field and procedure help to supplement your *User Guide* set. Printed field help is available through Field Help Report (36.4.14). The Procedure Help Report (36.4.16) prints procedure help in alphanumeric ranges by program name.

The Field Help Book Report (36.4.15) enables you to print a book containing all field help. Choose units as small as one field and as large as an entire module.

Local Vars. Set to No to exclude local variables. These are field names created within a program, not drawn from the data dictionary. In reports, the From and To fields are often local variables. Usually, help for local variables is not as significant as database fields.

Update Only. Set to Yes to limit output to fields that can be changed.

Where-Used, Maximum. Set Where-Used to No to keep the system from printing a where-used list after each help item. Some database fields are used throughout the database, and a complete where-used list can be very long. If Yes, limit the length of the where-used list by entering a value in Maximum.

Building an E-Mail System Interface

Some functions can be configured to send e-mail messages to designated users. For example, optional e-mail messaging is used in System Security, Product Change Control, Supplier Performance, and the Global Requisition System.

To take advantage of this feature, the e-mail system must be defined and addresses specified. The e-mail interface is built around an operating-system command that communicates with the user's e-mail system. This command tells the e-mail system how to construct and address messages.

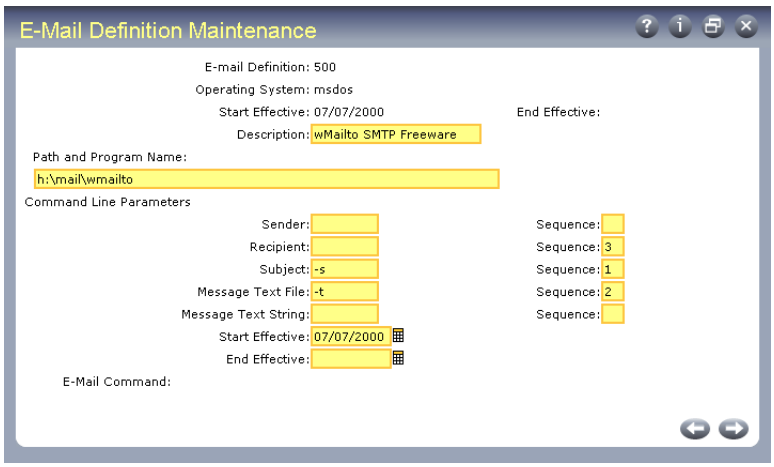
Set up a command line in E-Mail Definition Maintenance (36.4.20) for each system you want to access. Then, in User Maintenance (36.3.1), specify an e-mail definition and address for each user.

Be sure that an output device is defined in Printer Setup Maintenance (36.13.2) that has Destination Type set to Email. This is described in "Setting Up Printers" on page 57. When you select the associated device in the Output field in programs throughout the system, the resulting report is sent to specified e-mail addresses.

E-Mail Definition Maintenance

Before you implement E-Mail Definition Maintenance (36.4.20), refer to the e-mail application documentation or consult with your e-mail system administrator to determine if the application you are using provides an operating-system command interface. If it does not, various shareware products provide e-mail command-line interfaces.

Fig. 4.9
E-Mail Definition Maintenance (36.4.20)



E-Mail System. Enter an alphanumeric code for an e-mail system your company uses. This can be a number or a shortened version of the application name. You can use the same code for more than one record to give users access to multiple systems. For example, you can define both a UNIX system and a Windows system with the same code so that a user can log on to either system with the same user ID.

Operating System. Enter the name of the operating system on the user's computer. This is not necessarily the same operating system as the computer where the databases reside. Valid values are UNIX, MSDOS, and WIN32.

Start Effective. Optionally enter the first date this system is available for use.

Description. Enter a brief description of this system.

Path and Program Name. Enter the complete path to executable e-mail application file; for instance:

```
F:\apps\shared\email\blat.exe
```

End Effective. Enter the last date this system is available for use. This is an optional field.

Command line parameter fields can store parameters or arguments to identify the type of data being passed to the command. The parameter is a prefix, which is followed by the type of data. The UNIX `mailx` command, for instance, requires that the subject of the message have a -s prefix, as in the following example:

```
mailx -s "test message"
```

E-Mail Definition Maintenance defines four parameters: Sender, Recipient, Subject, and Message Text File (or Message Text String). Use the message parameters required by your e-mail system. Only one message field can be used in each e-mail definition.

The Sequence fields control the order in which the Sender, Recipient, Subject, and Message Text parameters appear in the command line. Some e-mail systems require these parameters in a specific order. If your system does not use one of the parameters, leaving both the Parameter and Sequence fields blank omits that parameter from the command line.

If you enter a parameter without a sequence, the parameter is not included on the command line. If you enter a sequence without a parameter, the system skips this parameter and creates the command.

The E-Mail Command field displays the system-built Path and Program Name, Parameters, and Sequence.

When you complete the setup for your e-mail system, you are prompted to send a test message. The default addressee is your log-on user ID. If you have not yet entered your e-mail address in User Maintenance, the system prompts you for an address.

User Maintenance

To use the e-mail interface, you must also complete two fields in User Maintenance (36.3.1) for each user: E-Mail Address and Definition.

E-Mail Address. Enter the complete e-mail address for this user, as required by your company's e-mail system.

E-Mail Definition. Enter a code established in E-Mail Definition Maintenance.

See "Defining Users" on page 155.

Using Advanced Reporting Tools

If you have the QAD .NET UI, use programs on the Report Setup Menu (36.4.21) to support advanced reports and dashboards designed using the Cognos reporting tool. Additionally, several QAD-designed dashboards are available with QAD Business Intelligence 2.5.

See *Technical Reference: QAD Business Intelligence* for detailed information on these programs.

Dashboards add an interactive element to reports. They let you:

- Drill up and down to see higher and lower levels of detail.
- Include multiple charts derived from different data sources in a single report.

Important Although the setup menu is available in all user interfaces, you can only view the resulting reports and dashboards through the QAD .NET UI.

QAD-Provided Dashboards

If you have purchased QAD Business Intelligence 2.5 and the appropriate supporting elements, you can implement several QAD-provided dashboards. See *Technical Reference: QAD Business Intelligence 2.5* for detailed requirements and procedures.

Custom Reports and Dashboards

You can implement custom reports and dashboards in without using QAD Business Intelligence, as long as you have installed the following components:

- QAD .NET UI
- QAD ReportNet Bundle, delivered with the QAD .NET UI
- Cognos 8.2

Use the following workflow to implement custom reports and dashboards.

1. Set up the QAD report server after installing Cognos 8.2.
2. Create reports and dashboards using Cognos Report Studio. See *User Guide: Cognos BI 8 Report Studio* for details.
3. Configure report settings and perform report synchronization using programs on the Report Setup Menu:
   a. Use Report Control (36.4.21.24) to configure report server settings and view or modify URL parameters.
   b. Use Report Synchronization (36.4.21.2) to synchronize reports between the system and the report server.
   c. Use Report Parameter Synchronization (36.4.21.4) to synchronize report parameters in the system with the report server.
4. Create menu entries for the new reports using Menu System Maintenance (36.4.4).

Chapter 5

Printers and Batch Processing

This chapter describes how to set up and use printers and batch processes.

Introduction

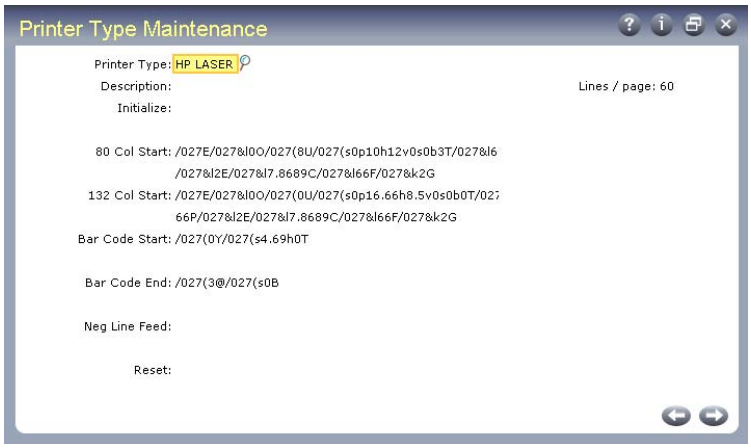
You can send reports, inquiries, and browses to a variety of printers—both local and network. The Printer Management menu contains programs for setting up system printers and default printers by user or group. Printers apply to all domains in a database.

The Batch Processing menu includes programs for creating batch print requests. You can edit and process batch jobs from multiple domains without having to switch the current working domain.

Defining Printer Types

Before setting up printers, define printer types using Printer Type Maintenance (36.13.1).

Fig. 5.1
Printer Type Maintenance (36.13.1)



Printer Type. Select your printer type from the list of predefined types. If your printer type is not in the list, use a similar printer type or define a new one.

To define a new printer type, you specify a series of programming sequences to control printer characteristics and behavior in the following situations:

- 80-character-width print jobs
- 132-character-width print jobs
- Barcode print jobs
- Hardware initialize and reset

Using control characters, you define how your printer performs such tasks as modifying fonts, changing page orientations, producing multiple copies, and so forth. Your printer manual is the best resource for control code definitions.

Note Without correct control codes, the related aspect of printer control will not work.

Use normal ASCII characters in the control fields. For nonprinting characters, also called control characters, use a slash and the three-digit ASCII number for the character. Table 5.1 lists characters frequently used in control sequences.

Table 5.1
Control Characters

Control Character	ASCII
Backspace	/008
Tab	/009
Linefeed	/010
Form Feed	/012
Carriage Return	/013
Escape	/027

Default system data includes correct control sequences for some commonly used printers.

Note One of the default printers is terminal. Use terminal in a character interface, window in a Windows interface, and page in QAD Desktop or .Net UI.

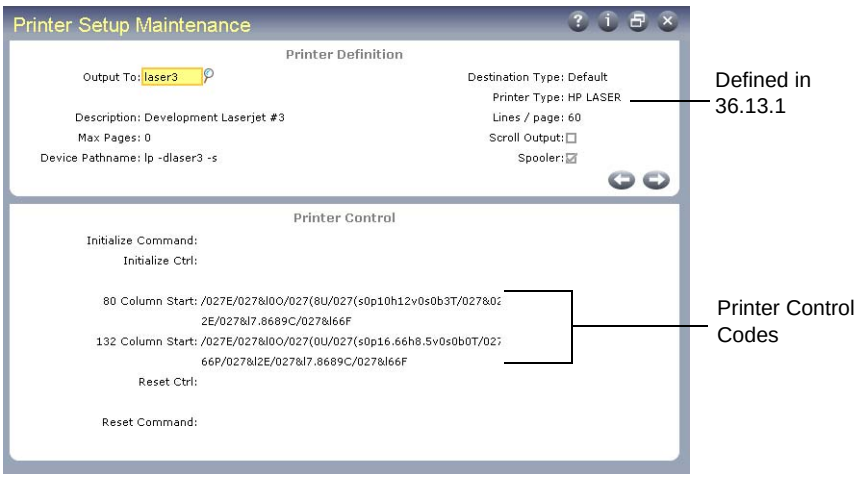
Table 5.2
Sample Printer Control Codes

Code	Function
/X27E	Printer reset
/X27&l3A	Folio paper format
/X27&lXO	Portrait orientation
/X27&l1O	Landscape orientation
/X27&l1S	Long edge binding (prints on both sides)
/X27&l66F	Bottom margin is 66 lines from top
/X27(sXp16.67h8.5vXsXbXT	Pitch 16.67, height 8.5, default style, thickness, font
/X27&l7X89C	Adjusts vertical index in steps of 1/48 inch
/X27(sXp16.67hXs3b4X99T	Pitch 16.67, height default, bold, courier (4X99)

Setting Up Printers

After you have defined printer types, use Printer Setup Maintenance (36.13.2) to set up printers and other output devices.

Fig. 5.2
Printer Setup Maintenance (36.13.2)



Output To. Assign a unique name to each printer or other output device. This name displays in the Output field of reports and inquiries. The QAD demo databases use *printer* and *terminal* for the most commonly used printers. However, you can use any name.

You can set up more than one record for the same printer, as long as you use different names in Output To. For example, this lets you access the same printer from both character and Windows clients.

Destination Type. Enter the type of device represented by this printer definition. Valid values are:

- Default. This is a server printer, a terminal display, a Windows display, or output to page. In Language Detail Maintenance (36.4.3), this mnemonic is assigned to value 0 (zero).
- EMail. This printer definition sends the report output to an e-mail message. For this to work properly, you must have an e-mail system that accepts a command-line interface. The e-mail system must be set up in E-mail Definition Maintenance, and the User Maintenance record for each user must include an e-mail definition and e-mail address. In Language Detail Maintenance, this mnemonic is assigned to value 1. See “Building an E-Mail System Interface” on page 51.
- Winprint. Use this type to represent printers selected from the Windows network of a GUI client computer. Devices defined with this type are available only from GUI clients. When you run a report and specify a Winprint device in the Output field, you can select a specific printer from your network and control some printing options through Windows dialog boxes. In Language Detail Maintenance, this mnemonic is assigned to value 2.

Printer Type. Optionally enter a printer type defined in Printer Type Maintenance. If you specify a type, the characteristics assigned to that type are copied into this printer setup record. You can modify them as required.

Description. Enter a description of the output device. Describing the physical location of a printer can be helpful.

Lines/Page. Enter the maximum number of lines to appear on a page. If you set up a printer to accept a maximum of 6 pages at 72 lines to a page, the printer prints only the first 432 lines of output, exclusive of the trailer.

Max Pages. Enter the number of pages a device can accept. If zero, no page limit applies.

Important System administrators should use Printer Setup Maintenance (36.13.2) to set a page limit of 1000 on the Output to Page option for reports. If you output a report of more than 1000 pages to Page, the retrieval of the data puts a burden on client resources and can cause system instability.

Note If you try to print checks, forms, and similar items on a device with a maximum page limit, an error message displays.

Scroll Output. Enter Yes to have the system accept a maximum of 3,000. Otherwise, the Max Pages limit applies.

Device Pathname. Specify the operating system command or path name that enables you to output to this printer. A device path name is normally not required for a terminal. However, if you are setting up a slave printer or a terminal window under X-windows, you may need to enter a path name. Table 5.3 lists examples of device path names.

Table 5.3
Sample Device Path Names

Device Path Name	Operating System	Effect
//arnt01/supjet1	Windows	Prints to network printer, shared as `supjet1` off the `arnt01` print server.
printer	Windows	Prints to Windows captured default printer.
lp -d supjet1	UNIX	Passes UNIX `-lp` command to operating system, causing printing at destination `supjet1.` Spooler must be Yes.

Spooler. Indicate if this is a spooled device. This field only applies to UNIX systems.

Initialize Ctrl/Reset Ctrl. A slave printer is one connected to a local PC printer port or the printer port of a dumb terminal. To transfer printer output to the proper port, you may need to specify control codes for these fields. The initialize control string passes output from the terminal to the print device. The last section of the Reset control string returns output to terminal. Set up control strings for each printer. In UNIX, the slave printer device path name is:

```
/device/tty
```

Defining a Printer for Use with Other Interfaces

If users generate reports from the QAD Desktop or .NET interface and want to view them immediately, they should choose the Page output device rather than terminal. Output to terminal is not formatted to display correctly in a browser.

The Page output device should be defined with the following settings:

- Max pages is 0.
- Destination type and printer type are blank.
- Lines per page is 66.
- Scroll output is Yes.
- Spooler is No.

Setting Default Printers

Use Printer Default Maintenance (36.13.4) to assign default output devices to users. This is only the default; you can change it to any valid device when you run the program. You can apply a record to all users by entering an asterisk (*) in the User ID field.

Note Default output devices apply only to reports; the default device for inquiries is always terminal.

You can specify devices for a user ID or a combination of user ID and menu selection. This can be useful for specialized tasks such as sending checks to a check printer; the same user can have different default output devices for different programs.

The default does not necessarily have to be a physical printer; you can also choose to send output to the terminal, page, a window (Windows UI only), or an e-mail recipient.

Defining Document Formats

Some programs let you specify alternative formats for printed documents in addition to the system-defined default formats. For example, an Italian customer may require a different sales order layout than a US customer. In that case, you can specify a predefined alternate format in the Form Code field of Sales Order Print (7.1.3).

You do not use a menu-level program to define alternate document formats. Instead, you must create a Progress program to generate them. Use the following steps to do this.

1. Create a Progress program to format the document as required.
2. Name the new program file appropriately so it can be located by the print program. The file name is typically created by removing the first two characters of the print program name and appending a two-character form code.
3. Modify the applicable print function to consider the new form code as valid.

Example You create two new sales order formats, identified with form codes AA and 2. The program name for Sales Order Print is `sosorp05.p` and the default sales order layout is defined by `sorp0501.p`. Use program file `sorp05AA.p` to store sales order form code AA and program file `sorp0502.p` to store form code 2. Be sure to include the zero preceding the 2. Then, modify `sosorp05.p` to define the two new formats as valid.

Running Batch Processes

A batch process is a group of processes run simultaneously. You can use batch functions to defer processing and report printing for reasons such as the following:

- A printer is busy or broken.
- Users want to be able to continue working without having to wait for lengthy reports to finish.
- Reports need to be run in a sequence, regardless of how they are submitted.
- You want to balance system load by running CPU-intensive programs when system load is low, perhaps at night.

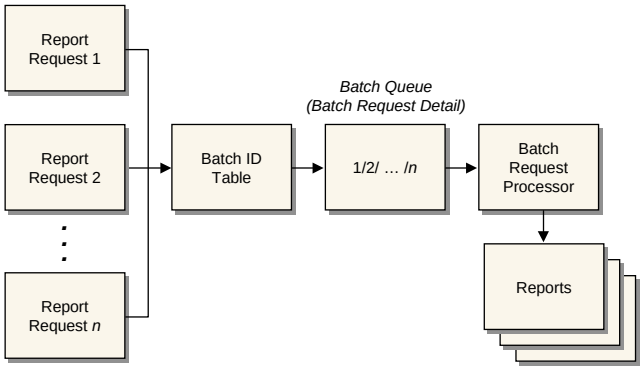
Define Batch IDs

To set up a batch process, system administrators first create batch IDs in Batch ID Maintenance (36.14.1). Use ID names that are descriptive and easy to remember, such as Paycheck, Monthly, or After5. You also assign the batch a priority that determines when it will run. Requests with the highest priority are run first.

Users then submit reports or programs that can be run in batch mode and specify the batch ID.

Note Batch IDs are domain specific. You must set up a separate set of IDs for each domain. You can, however, manage batch processes from multiple domains in the batch processing programs.

Fig. 5.3
Batch Processes



Review Batch Jobs

Usually the system administrator reviews batch requests prior to batch processing. Use Batch Request Detail Maintenance (36.14.3) to view reports and programs submitted to any batch. You can eliminate duplicate or unnecessary requests, prioritize requests, and redirect output as needed.

You must specify the domain associated with the batch requests you want to modify. You must have access to any domain you specify in User Maintenance (36.3.1).

As each request is executed, its status is updated to reflect whether it completed normally. Statuses include:

Failed (incomplete)

Complete

Running

When a batch does not complete normally, use Batch Request Detail Maintenance (36.14.3) to select the Process field and restart Batch Request Processor.

Process Batch Request

Use Batch Request Processor (36.14.13) to run reports and/or programs submitted by users with a batch ID. You can process up to 10 batch IDs in a single run. Each batch ID can be associated with a different domain. This lets you manage batch requests for multiple domains within one database without having to change your current working domain.

When you run a batch process, the system executes all items queued for a given batch ID in the requested order. You control the batch order by assigning a priority to each batch ID.

Invoke Batch Processing from CIM

In UNIX or Windows, you can create a batch file that invokes batch processing. You can then schedule when to run the script of the batch file. The scheduling capability of the operating system lets you run the batch processing at a time that is most convenient for you.

To set up a batch script, follow these steps.

1 Prepare a file that anticipates all data entry to Batch Request Processor (36.14.13).

The file should use CIM format. The first line provides login information:

```
"<User_Name>" "<Password>" "<Login_Domain>"
"mgbatch.p"
"<Domain>" "<Batch_ID>"
- - - - - - - - - - - - - - - - - - -
"<Is_Repeat>"
-
.
.
"Y"
```

In the script, `mgbatch.p` is the program name for Batch Request Processor. *`Domain`* and *`Batch_ID`* identify the batch requests to process. The line *`Is_Repeat`* indicates that requests for multiple domains can be included in the script. A hyphen (`-`) indicates to tab through a field; the two dots are exits, and `Y` confirms the exit from your session.

See Chapter 6, "CIM Interface," on page 65 for more details on CIM load processing.

2 Create a `.p` file of following format. Replace *`Input_File`* with the path of the file that you prepared in the previous step.

Note If you are working in UNIX instead of Windows, the first statement in the following script is unnecessary.

```
Assign PROPATH = <Propath>.
Input From <Input_File>.
Output To <Output_File>.
Run mf.p.
Input Close.
Output Close.
```

3 Set up a batch file. The batch file is a `.sh` file that can be scheduled using the UNIX `crontab` command or a `.bat` file that, in Windows, you can add to Scheduled Tasks in Control Panel.

To set up the batch file, use the Progress command `mpro` (UNIX) or `prowin32.exe` (Windows) to invoke the `.p` program that you created in step 2.

- In UNIX, the `.sh` file has the following structure:

```
TERM = <Term>;
DLC = <DLC>;
PATH = <Path>;

PROPATH = <Propath>;

mpro <DB_Parameters> -p <Progress_Program> <Startup_Parameters>
```

- In Windows, the `.bat` file has the following structure:

```
SET DLC = <DLC>

SET PATH = <Path>

prowin32.exe <DB_Parameters> -p <Progress_Program> <Startup_Parameters>
```

The table describes the variables used in the scripts.

Table 5.4
Variables in Batch File

Parameter	Description
DLC	Specify the value of the *DLC* system variable.
PATH	Specify the value of the *PATH* system variable.

Parameter	Description
TERM	For UNIX only, specify a terminal type.
PROPATH	Specify the value of the Progress *PROPATH* variable.
DB_Parameters	Specify the parameters to connect to the database. For more information, see Progress help.
Progress Program	Specify the path of the `.p` program that you created in the previous step.
Startup Parameters	Specify other parameters for `mpro` or `prowin32.exe` to start. For more information, see Progress help.

Chapter 6

CIM Interface

This chapter describes how to use programs to manage the movement and storage of data in a database.

Introduction

Transferring data can save disk space, increase disk access speeds by compacting fragmented data, and integrate legacy or otherwise noncompatible data with QAD data. There are three basic ways to transfer data into and out of your QAD database:

- Dump or load data files.
- Archive and delete or reload data files.
- CIM load data files.

The first two options are discussed in Chapter 7. This chapter discusses CIM data load, which lets you load data into the system from any source, as long as the data is formatted to match the schema.

See page 75.

CIM is typically used to add or modify records in a database. In certain cases, it can also be used to delete records. Only some functions support this feature.

See “Deleting Records through CIM” on page 72.

Unlike direct data loads, CIM checks load data for errors and saves unloaded records in an error file for correction and reloading. CIM loads can be run in either batch or continuous mode.

Note Q/LinQ offers more advanced features for data transfer, including methods similar to CIM. See *External Interface Guide: Q/LinQ*.

Using the CIM Interface

The CIM interface loads data through online maintenance programs. All data validation used in these programs during normal data entry is available during a CIM load. Imported data is then made available to other programs.

Most of the data loaded through CIM is loaded into a specific domain. The domain used is the one the user executing the CIM function is currently logged into. If you have access to multiple domains, make sure you are logged into the correct one before beginning the load.

In UNIX, use an external load program to load data continuously. These programs can accept input from devices such as barcode readers.

If data is loaded directly into tables using dump/load programs or Progress loads, some tables may not be updated correctly.

Load data into the system using functions on the CIM Interface Menu (36.15). Imported data can come from:

- Any ASCII file that follows the correct conventions.
- The output of programs that run in multiprocessing environments such as UNIX.

See “CIM Data Format” on page 68.

To load a product structure, for example, construct a file that matches the record structure in the product structure master (ps_mstr), then load data into that table. The CIM interface enables you to construct a file of input values for Product Structure Maintenance (13.5), and then validates all the data.

Internally, the CIM Interface operates in two stages:

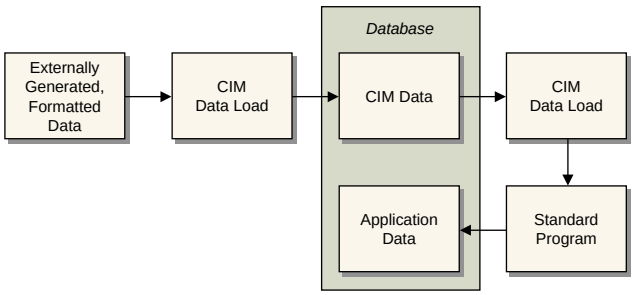
1. CIM Data Load (36.15.1) places data in CIM database tables. CIM Data Load can be executed as a Progress background session.
2. CIM Data Load Processor (36.15.2) sends data stored in CIM database tables through the appropriate input screen.

Both the data load and the data processor can be executed as a Progress background session.

Use other functions on the CIM menu to:

- Use CIM Data Load Process Monitor to monitor the load process, as needed.
- Use CIM Data Load Report/Delete to review processing errors and delete processed data, as needed.

Fig. 6.1
CIM Data Load

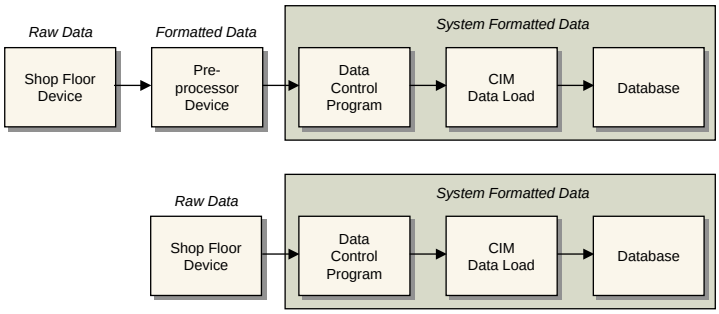


When CIM Data Load reads a data load group, it creates a record in the batch data load master table and assigns it a unique group ID. This integer record contains the name of the program to receive the data, and the date and time when the record was added. CIM Data Load then creates a record in the batch load detail table for each line of input data from the data load group.

Input from a file can be from either a disk file or a device-character file such as a serial port. If Input File/Continuous Process is selected, CIM Data Load executes the external program named in the Continuous Process Name field. The program controls and formats incoming data and sends its output back to CIM Data Load.

See “CIM Data Format” on page 68 for details.

Fig. 6.2
Continuous Data Input



Warning When acquiring external data in real time, run CIM Data Load at the highest possible dispatch priority to ensure that data loss does not occur as a result of competition with other system processes.

CIM Data Format

Each program takes in data in groups. A group typically consists of input fields within a frame. When using a program interactively, you must press Go to move from one group to another.

Data going into the CIM load must use the rules described in this section.

See “Determining Data for the Input File” on page 69.

The `@@BATCHLOAD` key word signals the beginning of the data-load group, consisting of one or more lines. Program name is the program that will process the input data. For example, if item data is being loaded, the program name would be `ppptmt04.p` (Item Data Maintenance, 1.4.3).

All input data contained between each `@@BATCHLOAD` and `@@END` is one group, regardless of how many transactions are specified in the data section.

Limit the number of transactions to 50. Each transaction entry can involve the creation of many records. The more transactions in a transaction group, the more system resources are required for processing, and the greater the likelihood of errors.

An error in one transaction can put all transactions in a group out of sequence and prevent the system from processing that group. In cases where maintaining data integrity is vital and re-creating data difficult, you might limit the number of transactions to one.

See “CIM Data Input File Example” on page 70.

Input File Formatting Rules

When creating your CIM input file, follow these formatting rules:

- Use a single line of data for each input request.
- To treat two consecutive input lines as a single line, place a tilde (~) at the end of the first line. Place no characters, including spaces, after the tilde.

  Note The tilde (~) is not required if you create the CIM file in an editor.
- Surround character fields with quotation marks.

- At the end of each input group, use a line feed. The end of an input line performs the same function as the Go key (F1 in character UI). Fields for which there are no data and that come at the end of an input sequence do not require hyphens.
- Type all characters in lowercase, taking care to spell correctly.
- Use a hyphen (-) to Tab through a field, retaining the default or existing value. For example, to accept default data for fields 1, 2, 3, 5, and 7, and enter Yes, 12, and 01/01/07 for fields 4, 6, and 8, enter the following:

  ```
  --- “yes” - “12” - “01/01/07”
  ```

- Format data as it is entered.
- Use a period on a line by itself to indicate End or End-Error.

  For repeated input (that is, multilevel), use the period to go back one level. This executes the End command.
- Use slashes (/) where needed. These are not required.
- Make sure the date format in the CIM file matches the date format specified in the Progress session startup parameters (`-d` parameter).
- Use a caret (^) to indicate a null value.

Input Data Types

Input data is information that you would normally enter from your terminal. The manner in which you enter information in an input file depends on the type of information the field is set up to handle. There are four types of input data:

- Character fields can be alphabetic or numeric but have no mathematical operations applied to them. Descriptions (alphabetic) and customer codes (numeric) are examples of character fields. Surround descriptions with double quotation marks (“ ”). The description is accepted without quotation marks, but may be interpreted as more than one input. If there is a space in the description, you must use quotation marks.
- Fields used in mathematical operations are numeric values. They can contain a decimal point (.) or a negative sign (–), but no other symbols, including commas (,) and dollar signs ($) are allowed. Do not use quotation marks for numeric values.
- Logical fields use Yes/No values and do not require quotation marks.
- Format date fields the way they are formatted in the source field.

Determining Data for the Input File

Each program contains one or more entry groups. Each entry group consists of one or more data entry fields in which data can be entered before pressing Go.

Example In Employee Maintenance (2.7.1) there are three entry groups, corresponding to the number of times you must press Go. Although direct correspondence between entry groups and frames is normal, it is not required. The three entry groups are:

- Key field group—employee code
- Address group
- Employee data group

Each entry group corresponds to one line in a CIM file.

While navigating a program to determine field groupings, use the Tab key to move from field to field, rather than the Return key. The Return key works like the Tab key in all fields except the last field in an entry group, where it executes the Go command. This can be misleading in determining which fields belong to an entry group.

CIM Data Input File Example

```
/* wocimp.p */
/* Program to create CIM input data file for Work Order Receipt Backflush */
DEFINE VARIABLE wonbr LIKE wo_nbr.
DEFINE VARIABLE wolot LIKE wo_lot.
DEFINE VARIABLE woqty LIKE wo_qty_comp.
DEFINE VARIABLE woyes AS LOGICAL INITIAL yes.
DEFINE VARIABLE wono AS LOGICAL INITIAL no.
DEFINE STREAM bf.
OUTPUT STREAM bf TO batchloa.d.
REPEAT:
         PROMPT FOR wonbr wolot woqty.
         wonbr = INPUT wonbr.
         wolot = INPUT wolot.
         woqty = INPUT woqty.
         /* See if work order exists in system. */
         FIND FIRST wo_mstr WHERE wo_nbr=wonbr AND wo_lot= wolot NO-LOCK NO-ERROR.
         IF AVAILABLE wo_mstr THEN DO:
                  /*Identify beginning of record & program used.*/
                  PUT STREAM bf “@@batchload wowoisrc.p” SKIP.
                  /*The work order number and ID.*/
                  EXPORT STREAM bf wonbr wolot.
                  /*qty comp., issue alloc-yes, issue pick-yes*/
         EXPORT STREAM bf woqty woyes woyes.
         /*Component issue - yes.*/
         PUT STREAM bf “.” SKIP.
         /*Display items being issued - no.*/
         PUT STREAM bf “.”.
         /*Is all information correct - yes. */
         EXPORT STREAM bf woyes.
         /* Qty complete. */
         EXPORT STREAM bf woqty.
         /* Remarks - no. */
         PUT STREAM bf “-” SKIP.
         /*Display item and lot/serial detail - no. */
         EXPORT STREAM bf wono.
         /*Is all information correct - yes. */
         EXPORT STREAM bf woyes.
         /* Please confirm update - yes. */
         EXPORT STREAM bf woyes.
         /* Identify end of record. */
         PUT STREAM bf “@@end” SKIP.
         END.
END.
OUTPUT STREAM bf CLOSE.
```

Creating a CIM Input File

To create a data input file, first determine the program to be used and fields to be updated. The basic steps are as follows:

1 Run the program that is to receive the data and determine the program name.

Note You can also run Menu System Report (36.4.5).

- **a** In the character interface, use the Ctrl+F key combination to display the program context, including name.
- **b** In the Windows interface, display the About screen from the Help menu.
- **c** In the Desktop interface, click the i (information) icon to display program details.

d In the .NET UI, right-click on the option in the menus and choose Properties to display program details.

2 Determine the program's key fields. These are typically the first fields, and always let you advance to the next field by pressing Go.

A good test is to position the cursor in a field, and press Go. Note where the cursor goes. Reposition the cursor in the field, and press Return. If the cursor moves to the same place as it did when using Go, embed Go (Carriage Return) in your CIM file. If the cursor went elsewhere, embed a Return. You could still embed Go if this new cursor position did not lead to any field you want to populate.

An input file must contain values for key fields, each on a line by itself. This allows the Go command to apply to the appropriate field.

Note which fields are validated or secured. Do this by typing any character (for example, x) and pressing Enter. If a warning displays, the field is validated or otherwise constrained. Your input file must conform to valid choices for the field. Use the look-up browse for a list of valid entries.

3 Choose non-key fields you want to populate and in what sequence. Note whether Go or Return is required after each entry.

Not all fields have labels. For example, a two-line description can consist of two separate fields. To determine which lines correspond to which fields, place the cursor in each line and press Ctrl+F to display their field names. You must populate each field with a separate entry in a CIM file.

Note In QAD Desktop and .NET UI, field names display as field tips.

4 Record a template of the CIM input file entries for the first frame.

The following is an example template for Item Master Maintenance (1.4.1):

```
@@BATCHLOAD ppptmt04.p
"10-10000"
"EA" "Oasis Cooling System" "Home/Indust Model"
```

Remember, all CIM files start with `@@BATCHLOAD <Program Name>`. The Item Number (10-10000) is a key field and is required. It must be on its own line. The second line represents the next three fields in the entry group.

Follow Item Number with Go. The next line fills in the UM and Description fields. Note that Description is shown as two entries, one populating the first line, one populating the second.

Note There are a few cases where CIM load does not work, such as costing data in Item Master Maintenance (1.4.1). In this case, costing data has to be CIM loaded through Item Element Cost Batch Load (1.4.15).

Use the following code to load this data.

```
@@batchload ppptmt04.p
“10-10000”
“Ea” “Oasis(TM) Cooling System” “Home/Indust Model”
“1000” “5/28/1992” “Config” “AC” “DISCRETE” “10-10000” “AB”
.
@@end
```

Error Handling

When the CIM load is completed, CIM Data Load Processor (36.15.2) creates a report showing the groups successfully processed and any processing errors. Groups containing an error are not processed. Troubleshoot errors using the following guidelines:

- Are the values appropriate?
- Is there a line reading: `@@batchload`?
- Is there a line reading: `@@end`?
- Are the data in the correct order?
- Are there any blank lines?
- Are there any misplaced spaces?
- Is there an end-of-line for each data set?
- Does it complete the record?
- Did the first error cause all the others?

Deleting Records through CIM

You can use CIM to delete records created with any of the programs listed in Table 6.1. In each of these programs, an updateable, single-character field, `batchdelete`, exists at the end of the header- and detail-record key frames. This field can be updated only when the program is accessed through a batch process, that is, when `batchrun = true`.

Note When you press Ctrl+F in a field of a program with batchdelete enabled, a pop-up window appears, indicating that you can use batch delete. You can do this in the character and Windows interfaces.

Table 6.1
Programs with `batchdelete` Functionality

Menu Label	Program Name
Customer Maintenance	adcsmt.p
Customer Ship-To Maintenance	adstmt.p
Customer Item Maintenance	ppcpmt.p
Generalized Codes Maintenance	mgcodemt.p
Site Maintenance	icsimt.p
Entity Code Maintenance	glenmt.p
Account Code Maintenance	glacmt.p
Sub-Account Code Maintenance	glsbmt.p
Cost Center Code Maintenance	glccmt.p
Currency Maintenance	mccumt.p
Price List Maintenance	pppimt.p
Price List Maintenance	pppcmt.p
Item Master Maintenance	ppptmt.p
Installed Base Item Maintenance	fsisbmt.p

Because the batchdelete value exists at the end of key frames, it does not affect existing CIM input files and can be omitted from these files when not used. Since it is only one character, unlabeled, and hidden, the field also does not change the visible interface.

Creating Input Files to Delete Records

Use these guidelines when creating input files that include deletes:

1. To determine if batchdelete is enabled in a particular program, check the list in Table 6.1.

   Note In the character and Windows interfaces, press Ctrl+F to display the information window. It indicates whether batch delete is available.

2. To invoke the batch delete functionality, place an x at the end of the header- or detail-record key frame line in the input file.

3. Follow the key frame with a blank line consisting of a single hyphen so that the program executes the code that would be executed if an F5 or Ctrl+D has been pressed in the first frame after the key frame.

4. Enter a subsequent line containing the string yes as an answer to the Please Confirm Delete prompt displayed for online deletes.

Example of CIM Delete

The first CIM input file creates a GL sub-account. The next two input files use the delete functionality first to delete one sub-account line then to delete the entire sub-account record.

Add or modify a GL sub-account record with three lines.

```
@@BATCHLOAD glsbmt.p
sbtest
“test sub-account”-
1
1040 1041
2
1050 1051
3
1060 1061
@@END
```

Delete the second sub-account line. The detail-record key frame for the second line ends with x, followed by a blank line, and yes confirming the deletion.

```
@@BATCHLOAD glsbmt.p
sbtest
--
2 x
-
yes
@@END
```

Delete the entire GL sub-account record with all of its lines. The header-record key frame ends with x, there is a subsequent blank line, and yes to confirm the deletion.

```
@@BATCHLOAD glsbmt.p
sbtest x
-
yes
@@END
```

Running Multiple CIM Sessions

Any number of CIM sessions can be run at one time. However, two load sessions cannot be opened for a single file. To run two sessions, divide the file.

When running multiple sessions, use CIM Data Load Process Monitor (36.15.4). The monitor shows the state of all existing CIM sessions. Type and Process Session are indexes to the sessions. Enter Process in Type and use (/) to first see all the Process sessions, followed by the Load sessions. If you select Go at the Session field, the current status of the processes displays continuously. The display shows startup time, last transaction time, and selection criteria used when the session was started.

Killing CIM Sessions

Although a CIM session runs under the operating system and can be stopped using operating system commands, this is not advised. When the operating system kills a session, the user is not notified and a record of the session may still display in the CIM Data Load Process Monitor (36.15.4).

The best way to kill a CIM session is to use the Process Monitor. To kill a session, identify the session using the Type and Session fields then press the F5 key in the Session field. A prompt asks you to confirm that you want to delete this record.

If the session was invoked with a low-dispatch priority, your monitor may still display a session after it has been stopped, with a status of Killed. To erase the session from the system, delete it again by putting the cursor on the Session field and pressing F5.

Chapter 7

Database Management

The system provides utilities for monitoring database size, performing dumps and loads, reloading archive files, managing database sequences, registering applications, and monitoring license compliance.

Managing Database Size

Several utilities help you manage the size of your database.

Determining Disk Usage

Use Database Table Size Inquiry (36.16.1) to dump selected tables and review their sizes. Reported table sizes may be understated since indexing overhead is not taken into account.

Note The program requires adequate free disk space to run.

Use Disk Space Inquiry (36.22.13) to display free space for each available disk, in blocks. For most UNIX environments, a block is typically 1024 bytes. For Windows environments, blocks range from 1024 to 8192 bytes. Consult your hardware manuals for exact specifications.

Note These programs must be run from a character user interface.

Fig. 7.1
Disk Space Inquiry (36.22.13)

```
/                 (/dev/vx/dsk/rootvol):  956656 blocks   464665 files
/proc             (/proc             ):       0 blocks     4453 files
/dev/fd           (fd                ):       0 blocks        0 files
/tmp              (swap              ): 7823264 blocks   381700 files
/opt2            (/dev/vx/dsk/crsu03_dg/vol04): 2757000 blocks   948168 files
/dr01            (/dev/vx/dsk/crsu03_dg/vol01):46291736 blocks 12355240 files
/dr02            (/dev/vx/dsk/crsu03_dg/vol02):48571390 blocks 12427225 files
/dr03            (/dev/vx/dsk/crsu03_dg/vol05): 9841572 blocks  2461436 files
/opt.new         (/dev/vx/dsk/crsu03_dg/vol03): 8622328 blocks  2448537 files
/users/cmb       (qcrhp01:/disks/drive2/d7/users/cmb):  654480 blocks      -1
 files
/users/dzn        (qcrhp06:/dr4/users/dzn):  422860 blocks       -1 files
/users/svc        (ohhp04:/home/u3/svc): 1401846 blocks       -1 files
/users/fxd        (ohhp04:/home/u3/fxd): 1401846 blocks       -1 files
/users/pzd        (ohhp04:/home/u3/pzd): 1401846 blocks       -1 files
/users/byd       (qcrhp01:/disks/drive2/d7/users/byd):  654480 blocks      -1
 files
/users/rbe       (qcrhp01:/disks/drive2/d7/users/rbe):  654480 blocks      -1
 files
/qad/mfgpro/85db/etfdb(ohhp40:/dr01/85db/etfdb): 9285970 blocks       -1 files
/users/svb        (ohhp04:/home/u3/svb): 1401846 blocks       -1 files
/users/ncr        (ohhp04:/home/u3/ncr): 1401846 blocks       -1 files
/users/scq        (qcrhp06:/dr5/users/scq): 3373932 blocks       -1 files
```

Freeing Disk Space

There are three ways to reduce the size of a Progress database:

- Use dump/load programs to compact your data. Compacting data can increase disk access speeds significantly. To do this, dump all data from your database, and reload it into an empty database. You need free disk space amounting to about 70% of the total size of your data (`.d`) files. Progress recommends that you dump/load once a year.
- Use delete/archive programs to create free database space. Typically, the largest tables in a database contain history, sales order, and purchase order data. The amount of disk space may decrease if you store the archived data on the same disk. See “Deleting and Archiving Data” on page 78.
- Use both dump/load and archive/delete programs. To do this, remove records from the database, dump the remaining data, and reload it into an empty database. You need plenty of free disk space to do this.

Dumping and Loading Data

Dump/load programs move the contents of database tables into or out of ASCII files. The dump procedure reads a database table, puts quotation marks around the data value of each field, and places those values in an ASCII file.

Example A record in the user master table (usr_mstr) consists of the following entries:

usr_lang	FR
usr_site	1000
usr_user1	
usr_user2	
usr_user ID	pxr

One line in the dump file would read:

```
“FR” “1000” “”””””pxr”
```

You can use dump files as input to other programs after converting the files to CIM input-file format. You can also take output from other programs, convert it to CIM input-file format, and load it into the database. This assumes the data has the correct form, based on the screen flow and format the CIM input is duplicating. The *Database Definitions* book contains details on specific table formats.

See “Using the CIM Interface” on page 66 for details.

Dump/load procedures are located at 36.16.4 in the Windows interface and at 36.16.3 for UNIX environments. Load procedures do not overwrite existing records. You must delete the old data first.

Note Progress and Oracle each provide dump/load and import/export programs, but these programs do not maintain the integrity of data in the database. For information on Progress dump/load and bulk load programs, see the Progress user manuals.

Dump/Load Procedures

To dump/load data:

1. Back up the existing database.
2. Check available disk space. A full dump/load requires free space equaling approximately 70% of existing database size. See “Determining Disk Usage” on page 76.
3. Log in to the system in single-user mode. You can speed up the dump/load by running multiple sessions of Database Table Dump/Load from multiple terminals.
4. Execute Database Table Dump/Load for the correct range of tables.

   If there is enough free space, select all tables. If there is not, archive the dumped files to a tape, then erase them from the database. Repeat this step as needed.
5. When the dump is finished, copy the standard, empty database (mfg) onto your old database.
6. Load the dumped files back into the database using Database Table Dump/Load.

Data files (.d files) reloaded into databases containing data do not overwrite existing records. Files to be loaded must be in a directory specified in your PROPATH. A Progress bulk load is usually faster than a dump/load, but can require an index rebuild.

The system lists load errors in a .e file located in the directory you ran the process from.

Deleting and Archiving Data

Delete/archive programs remove selected records from the database, letting you archive them to tape or other media. Each delete/archive screen looks similar to a report criteria input screen. You choose records based on selection criteria. Criteria can include date ranges, document numbers, employee names, and so on.

Table 7.1 lists data that can be deleted and archived.

Table 7.1
Transactions that Can Be Deleted/Archived

Accounts Payable	GL Transactions	Repetitive History
Accounts Receivable	Inbound EDI Documents	Retired Fixed Assets
Audit Detail	Installed Base History	RMA History
Call/Quote History	Intersite Requests	Routings
Closed Cumulative Orders	Intrastat History	Sales Analysis
Closed Intersite Demand	Invoice History	Sales Order Shippers
Closed PO Shippers	Kanban Transactions	Self Bills
Closed Projects	Logistics Charges	Sequence
Closed Purchase Requisitions	Lot Masters	Service Contracts
Closed Purchase Orders	Master Bills of Lading	Service/Repair Orders
Closed Purchase Receipts	NRM Sequences	Shippers
Closed Service Requests	Operation History	Subcontract Shippers
Comment Cross-References	Operation Plans	Supplier Performance Data
Containers	Operation Plan Simulations	Supplier Schedules
Customer Schedules	Outbound EDI Documents	Transaction History
Deferred/Accrued Revenue	Physical Inventory Tags	Turnaround Data
Expired Sales Quotes	Product Change Orders	Uninvoiced Receipts
Expired Call Quotes	Product Change Requests	WIP Lots
Family Hierarchies	Product Structures	
Flow Schedules	Q/LinQ Documents	
Forecast Details	Quality Orders	
GL Report Images	Quality Test Results	

Audit Detail Delete/Archive

Use Audit Detail Delete/Archive (36.23.1) to delete/archive audit detail information. Unlike other delete/archive programs, this program does not delete each record specified. Instead, for each unique combination of user ID, table, and field, it keeps the latest record and deletes/archives the rest.

To delete and/or archive tables:

1 **Back up your database and `.df` files**.

To safeguard against data archived from a previous product version that has different schema, back up the current database definitions (`.df`) file with each archive/delete run. This lets you reconstruct a corresponding database for data retrieval.

2 Verify record selection.

Run the delete/archive program without deleting or archiving records. This generates a report showing selected records. Review the report and if records selected for deletion are correct, proceed with the actual archive/delete.

3 Run appropriate historical reports such as Invoice History Delete/Archive (7.13.23).

4 Determine selection criteria for the records being deleted, and run the delete/archive program, setting Delete and Archive to Yes.

The program creates a `xxyymmdd.hst` file in the default directory where xx is the record identifier, such as iv for invoices, and yymmdd is the archive date.

5 Verify deletion of records from the database.

6 Verify the contents of the `.hst` file using the appropriate operating system command.

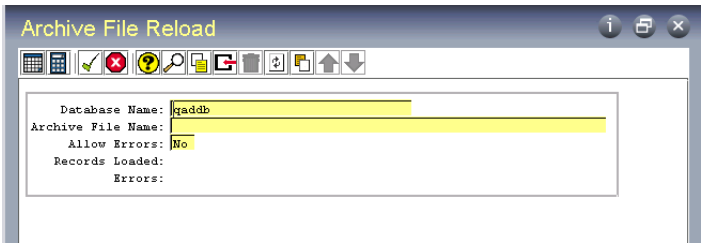
7 Back up the `.hst` file to storage media and delete from system.

The delete/archive program does not reduce database size. To reduce database size, use a dump/load program.

Restoring Archive Files

Use Archive File Reload (36.16.5) to reload an archive file after restoring the file from backup media to the system disk.

Fig. 7.2
Archive File Reload (36.16.5)



The reload process puts data from the archive file back into the database exactly as it was when you deleted it. However, if base data has changed, you may encounter errors.

Example You are reloading accounts receivable history for a customer that has been deleted.

Set Allow Errors to Yes to continue processing when errors occur. The system lists load errors in a `.e` file located in the directory you ran the process from.

Important Date and time in the stored data are formatted based on the country code associated with the user who archived the data. If a user with a different date and time format reloads the data, load errors and corrupted data can occur.

To avoid these problems, use the same user settings when archiving and reloading the data. Before loading data, use User Maintenance (36.3.1) to temporarily change your country code to match that of the user who archived the data.

See “Defining Users” on page 155.

Managing Database Sequences

When a unique identifier is needed by a program, the system often uses a control field to store the last number used. The system also supports the use of a special schema element called a sequence.

A *sequence* is a database element used to generate a stream of sequential values for assigning unique identifiers to records. Sequences allow fast, accurate numbering, and reduce the amount of time the system spends validating uniqueness.

Note Because the sequence is generated at the database level, records viewed from within a domain may appear to have gaps.

Use Sequence Report (36.16.15) to display a list of sequences defined in the database. The sequence description indicates the database table and field that is updated by the sequence. For example, the description of sequence cmt_sq01 is cmt_det.cmt_indx.

Sequences have the important advantage of speed and reducing the possibility of record locking and contention. However, each sequence is a separate database element, distinct from the table to which it applies. This means that sequences must be initialized correctly whenever you use Database Table Dump/Load.

If sequences are not initialized correctly, Duplicate Unique Key errors may occur when users attempt to create transactions.

If dumping and loading are done as part of installing a software upgrade, sequence initialization is automatically performed by the installation utilities. However, if you perform a dump/load to consolidate tables or increase database size, you must initialize sequences yourself. This is true also if you consolidate data from two different databases.

- Use Database Sequence Initialization (36.16.17) to reset sequences to the highest value plus 1 after loading data.This program works with both Progress and Oracle databases.
- Use Sequence Maintenance (36.16.13) to manually reset a sequence number to a specific value in a Progress database.
- Use Sequence Inquiry (36.16.14) or Sequence Report (36.16.15) to view sequence information.

To guarantee database integrity, perform sequence maintenance:

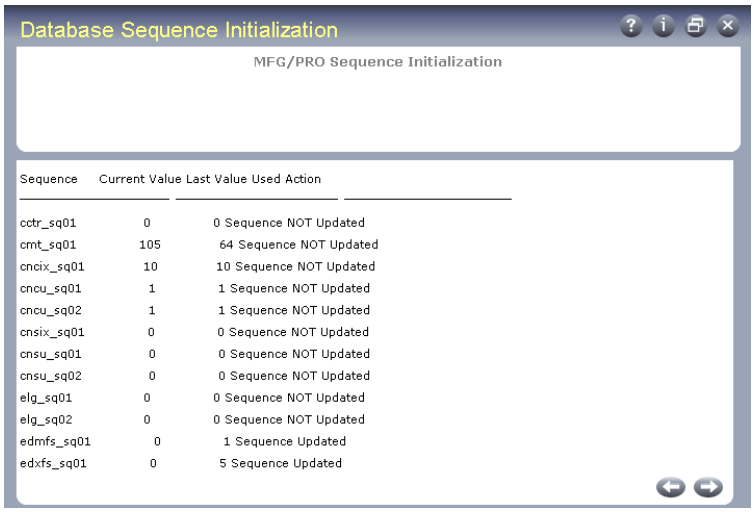
- In single-user mode sessions only
- As a required part of your standard database maintenance

Note To avoid accidental update to sequence structures, use menu security to protect sequence maintenance functions.

Initializing Sequences

Database Sequence Initialization reads each table that uses sequences and sets the sequence number value to the highest number plus 1. This ensures that each new record created has a unique number. This utility initializes sequences correctly in both Progress and Oracle databases.

Fig. 7.3
Database Sequence Initialization (36.16.17)



Maintaining Sequences Manually

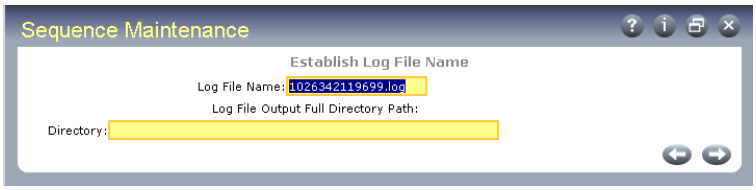
Maintain sequences manually or through the CIM interface. Maintenance includes:

- Dumping—outputting the current sequence value to a file
- Loading—reading a sequence value from a file
- Updating—manually updating a single sequence

See “Maintaining Sequences in Oracle” on page 84.

Maintain sequences in Sequence Maintenance (36.16.13). Sequence Maintenance works with Progress Relational Database Management System (RDBMS) only. Oracle dataservers are not currently supported.

Fig. 7.4
Sequence Maintenance (36.16.13), Establish Log File Name Frame



Log File Name. The name of the error log file.

Directory. The operating system (OS) directory where you want to store the file.

A second Sequence Maintenance screen displays.

Fig. 7.5
Sequence Maintenance, Sequence Name Frame

Sequence Name. Specify the sequence or set of sequences to be maintained. Leave blank to specify all sequences.

Note A time stamp is added to the log at the beginning of each session, so session history can accumulate. After a maintenance session, check the log for errors.

Maintenance Activity. Specify the maintenance activity to be applied to the specified sequence sets. Valid values are:

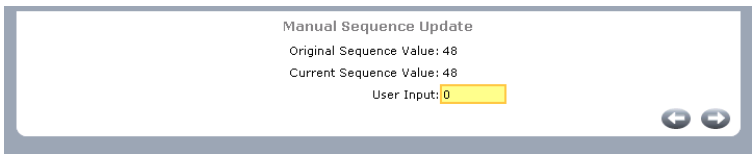
- 1 to dump. Outputs the current sequence value to an OS file.
- 2 to load. Reads the sequence value from the OS file.
- 3 to manually update. This activity can only be performed when a single sequence is specified. When a set of sequences is to be manually updated, the manual update activity is called once for each.

Activity Directory. For a dump or load, specify the OS directory where the sequence files are located. The direction of the data flow is determined by the activity.

Files are named using the name of the sequence with the file extension `.d`. For example, the sequence `tr_sq01` is dumped to a file named `tr_sq01.d`.

When a manual update is specified, an additional frame appears.

Fig. 7.6
Manual Sequence Update Frame



Original Sequence Value. This field displays the value of the sequence before the user's update was applied.

Note Sequence Maintenance generates a report listing current values of all sequences in the database. It can be run at any time and does not impact the content of sequence structures.

Current Sequence Value. This field displays the current sequence value.

User Input. Enter any sequence value within the valid range. The valid range is determined by the system and is part of the schema. An error displays when the value entered is not within the valid range.

Maintaining Sequences Using CIM

Sequences can be maintained using the CIM interface. The content of a sequence represents the last value applied to the sequence by a call from a function. This value is not available for processing, since it was consumed by another process.

Values used to update a sequence are validated against a range of acceptable values for the sequence. The value of the sequence can be within and including the boundary values. You receive an error message when the range is exceeded.

For more information on CIM, see “Using the CIM Interface” on page 66.

Limitations of CIM

Some limitations to maintaining sequences through the CIM interface are:

- Sequence maintenance must be performed in a single-user mode Progress session. The integrity of the sequence value is not guaranteed if maintenance is done in multiple-user mode.
- Destructive updates are not permitted. A CIM update cannot overwrite previously created files. Data dumping does not proceed if any elements in the set of sequences conflict with an existing OS file.
- You cannot manually update from CIM. CIM is an automatic process.
- Any error causes the sequence maintenance to fail. When you suspect a sequence maintenance activity failed while processing, you must repeat the entire process. This guarantees that the sequence values are valid.

Sample CIM File Format

A typical CIM file might look like the following example:

```
Line 1: <log file> <log directory>
Line 2: <sequence name>
Line 3: <action>
Line 4: <input-output OS directory>
```

<log file>. The name of the file receiving the output log. When an existing log file is specified, the current CIM output is appended to the end of the existing log. The default value is the value of the `mfguser` variable. This has the format of `TMP9999` where `9999` is a four-digit number that uniquely identifies the session. If the `mfguser` value is NULL (`""`), the log file is named `mgsqmt03`.

<log directory>. The location where the log file is stored. The blank value NULL (`""`) is specified as the default. When a `<log directory>` is not specified, the `<log file>` is placed in the PROPATH.

<sequence name>. Specifies the set of sequences to be maintained. You can specify a single sequence or the entire set. The default value is NULL (`""`), indicating all sequences will be maintained.

<action>. Specifies the activity to be performed, either (1) dumping or (2) loading.

Note The default activity is dumping (1).

<input-output OS directory>. The directory in which the sequence files are maintained. The default value is the local directory.

A time stamp is issued to the log file at the beginning of each session. This permits the same log file to accumulate a history of the session logs. All log files have the `.log` suffix.

Example The following is an example of a working CIM file:

```
@@batchload mgsqmt01.p
“sq_err.log” “/qad”
-
2
“/qad/backup”
@@end
```

This file outputs the error log to the directory `/qad` with the name `sq_err.log`. All sequences are maintained. The hyphen (–) indicates that the default value, in this case `all sequences`, is accepted. Number two (2) indicates that the sequences are loaded. The directory in which the sequence files are maintained is `/qad/backup`.

Note Only sequences currently implemented in the database can be maintained using CIM.

Maintaining Audit Trails

The system maintains an audit trail for all updates made to sequences using sequence maintenance routines. Each sequence has a separate set of audit entries.

For each updated sequence, the audit trail records original and final values. If the current value is the same as the original value, the system creates only one record.

Maintaining Sequences in Oracle

Normally, you use Database Sequence Initialization to set the starting sequence values in an Oracle database. The following information is provided if you need to manually maintain sequence values in Oracle, which cannot be done using Sequence Maintenance.

The standard sequence definition in Oracle is:

```
CREATE SEQUENCE <sequence name> START WITH <initial value>
INCREMENT BY 1 CACHE 75
```

Where `<sequence name>` is the same as defined in the Progress `df` and `<initial value>` is the starting value specified by the customer.

The initial value of a sequence is set to the highest value found in the field related to the sequence. The content of a sequence is the last value applied by a function.

See “Maintaining Sequences Manually” on page 81.

Example In a database with no user transaction processing, the maximum value of `tr_hist.tr_trnbr` is 1010. This value is used as the starting value of the sequence.

As user `qad`, you would enter the following SQL:

```
DROP SEQUENCE tr_sq01;
CREATE SEQUENCE tr_sq01 START WITH 1010 INCREMENT BY 1
   CACHE 75;
```

Registering Licenses

When you receive your software, you also receive license codes. This includes license codes for the foundation functionality and other separately licensed applications.

The license codes identify the license type, version, expiration date and number of days remaining, and number of users, sessions, or locations for which your site is licensed. Before you can use the system, you must register the license codes.

License registration programs are provided under the License Registration menu (36.16.10). Use the license registration programs to:

- Register newly installed software.
- Upgrade software to add new users or sessions.
- Maintain and report historical license data.
- Report detailed and summary license violations.
- Report license usage and user activity for QAD-conducted audits.

Licensing Overview

QAD licenses the software to its customers for use by a predetermined number of users, sessions, or transactions.

The following sections describe concepts associated with license types, user and location counting, license violations, violation types, violation messages, and registration interaction with other modules.

You can use User Monitor Inquiry (36.16.12) or other license-related reports to monitor user activities and application use.

License Types

Two license types apply to users:

Named User. Each unique user ID defined in User Maintenance (36.3.1) is counted as a user. There is no limit on the number of sessions each defined user can run simultaneously. Multiple sessions for the same user ID are counted as one user.

Concurrent Session. Each concurrent log-in is counted as a session. If a single user logs into multiple sessions simultaneously, each log-in is counted.

See “Violation Types” on page 86.

User Counts

The system monitors license use regardless of your user interface type, database type (Progress or Oracle), or license type.

For concurrent session license types, the system counts the number of active sessions when you log in and compares the count to the number of licensed sessions stipulated by the license agreement.

If you change to a domain in a different database, this process is repeated. This is because changing databases is like exiting your current database and starting a new session. Whenever you switch databases, the system stores the logout date and time.

Note If you use QAD Desktop or the .NET UI, each time you run a program and detach it in a separate window, each window counts as an individual session.

For named user license types, the software counts users when system administrators create new users in User Maintenance (36.3.1) or activate user access to applications in License Registration (36.16.10.1).

For location license types, the system counts the number of user locations and compares the number against the predefined limit for the license type when system administrators assign users to applications in either User Maintenance or License Registration.

See “Violation Messages” on page 87.

License Violations

When the number of users or sessions exceeds the amount stipulated by your license agreement, license violations occur.

The system stores all license violation occurrences. System administrators and QAD auditors can run reports to view the violation data.

The system responds to license violations with either violation errors or violation warnings. With errors, messages display and the system prevents additional users or sessions. With warnings, messages display, but additional users or sessions can exist and users can still log in to QAD applications.

See “License Reporting” on page 90.

System administrators can implement enforcement of license agreement by setting the Enforce Licensed User Count field to Yes in Security Control (36.3.24). Setting this field determines whether errors or warnings display and what action the system takes.

Important The first time a warning displays, you can access the system to complete transactions or other processing. If you receive repeated warnings, contact your QAD sales representative or distributor to upgrade your license.

The system prevents users from logging in if the license registration record does not exist. System administrators register the license code in License Registration (36.16.10.1).

Violation Types

The system records the violation types listed in Table 7.2.

Table 7.2
License Violation Types

Violation Type	Description
Date expiry	Displays information about violations that occur when an application's license registration expires. Only evaluation, demo, or temporary licenses have expiration dates.
Application Usage	Displays information about violations that occur when users do not have access to an application.
License Count	Displays information about violations that occur when the number of users or sessions exceeds the amount stipulated by the license agreement.
Non-Licensed Product	Displays information about violations that occur when users attempt to run applications that are not registered.

Violation Messages

Table 7.3 lists error messages that display when license violations occur.

Table 7.3
License Violation Error Messages

Message	Explanation and Solution
Expired license code	The license code expiration date for this application has passed. Contact your QAD sales representative or distributor to obtain a new license code. Register new code in License Registration (36.16.10.1).
Product registration is not valid	The licence code data in your environment has been corrupted or is missing. Contact your QAD sales representative or distributor to obtain the correct license code; register correct code in License Registration.
Application not available in licensed application master	Your environment license data has been corrupted or is missing. Contact customer support to reload valid license data.
Licensed user limit exceed	This message displays in User Maintenance and License Registration when the number of users exceeds the number specified by the license. System administrators can deactivate some users; otherwise, contact your QAD representative or distributor to upgrade your license agreement.
Customer is not licensed to execute this module/product: #	You selected a menu item that is not covered by registered license codes. Contact your system administrator to determine the correct menu items for you to access. System administrators should contact their QAD representative or distributor if the license code is not correct or if they wish to purchase this additional module.
User not authorized to run this application: #	You have not been authorized to run this product. System administrators authorize users to use products in User Maintenance or License Registration.

Message	Explanation and Solution
This product expires in # days on #	The license code for this application expires in the number of days indicated. Contact your QAD sales representative or distributor to obtain a new license code; register correct code in License Registration.
Concurrent session limit exceeded	The application you are attempting to access has a concurrent session license type and the maximum number of active sessions for this application has been reached. If this error displays during log-in, you cannot log in unless another currently logged-in user logs out.

Interaction with Other System Data

The license registration programs use data from other programs to process, maintain, and report license data.

System administrators maintain defined named users and a list of registered software applications that users are authorized to access in User Maintenance (36.3.1). License registration software uses this information to prevent more active users or locations than the license allows.

User Maintenance also includes information that more clearly defines the user. The system ships with a default set of user types predefined in Language Detail Maintenance (36.4.3). The set includes the employee, customer, and QAD user type. It is important for user count and system monitoring purposes that users are correctly identified in User Maintenance before complete license registration functionality can be used.

See “Language Detail Maintenance” on page 45.

License Registration

Use License Registration (36.16.10.1) to:

- Add a new license code for the base QAD ERP application or separately licensed modules.
- Upgrade license codes to add sessions, users, or locations.
- Remove license codes.

Fig. 7.7
Licensed Registration (36.16.10.1)

The system requests licensing information after you install the base product or separately sold modules and when you attempt to log in with an expired license.

Use the Tab key to select a license code task:

Add. The Add Product frame displays. Enter the license code for base QAD Enterprise Applications or a separately licensed module; then choose OK.

The application name, description, version, license type, and number of licensed users display.

When you add a license code, you are prompted to enter the IDs of users who can access the application. A list of users who can access the application displays once you enter a user ID.

See “Granting Users Access to Registered Software” on page 89.

If you try to add an application that is already registered, the following message displays:

```
Product already installed
```

Edit. The Edit Product frame displays. Use this frame to upgrade your license to increase users or sessions.You must obtain the new number from your QAD representative or distributor.

After you enter the code and choose OK, you are prompted to enter the IDs of users who can access the application.

Remove. The Remove Product frame displays. Enter the license code for the application you want to remove from registration. A prompt displays, asking you to confirm the license removal. If you select Yes, the system records the removal date and time. The application is no longer registered, and users cannot execute any programs that are a part of it. If you remove the license code, you will be logged out of the system, and users cannot log in.

Granting Users Access to Registered Software

You must grant users access to registered software. If a user who does not have access tries to start an application, either an error or warning message displays depending on the value of Enforce Licensed User Count in Security Control (36.3.24).

Access to applications is granted in one of two ways:

1. Assign access to individual users by selecting registered applications in the Application List frame in User Maintenance (36.3.1). See “Specifying Application Use” on page 163.
2. Activate users for a newly registered application in License Registration (36.16.10.1).

After you successfully enter a license code in the Add Product or Edit Product frames, the system displays the Add Authorized Users frame.

Fig. 7.8
License Registration,
Add Authorized Users Frame

User. Use this field to select users to be given application access:

User ID: Enter the user ID of the person you want to access the newly registered application. The User Selector frame displays a list, starting from the user ID you entered to the last user ID.

In the list, select the IDs of users you want to activate. An asterisk (*) displays on the left side of the user ID to indicate that the user is active.

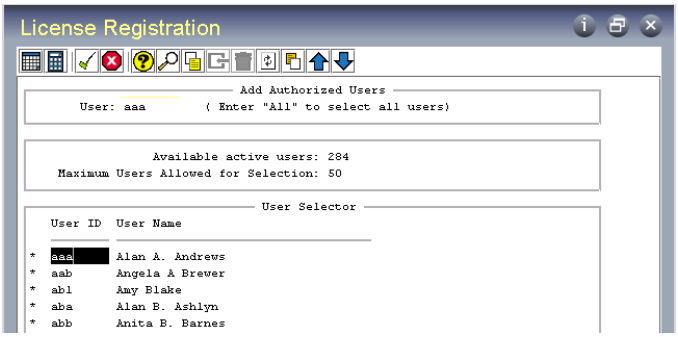
All: Enter the word All. The User Selector frame displays a list of all users. An asterisk displays on the left side of all users in the list.

To de-activate users, press Spacebar. The asterisk is removed.

Note If the total number of users exceeds the number allowed by the application's license, the system makes the first users in the list active. For example, if there are 100 user IDs displayed, but the license agreement for the application is for 50 users, the first 50 users are made active for the application.

If you need to authorize more users than your license allows, system administrators can add users through User Maintenance (36.3.1); however, the software records a violation of your license when you add more users.

Fig. 7.9
License Registration,
User Selector Frame



License Reporting

Various reports let you monitor application use, the number of logged-in users and sessions and the programs they use, and license violations. You can use the application usage and user count reports to be informed about potential license violations.

In addition to license reporting, you can use User Access by Application Inquiry (36.3.22) to display a list of applications, user access status (active or inactive), and access activation date.

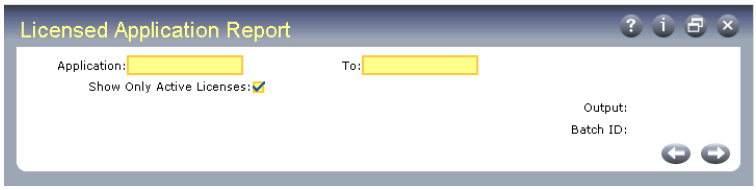
Licensed Application Report

Use Licensed Application Report (36.16.10.3) to display a list of software applications registered in the database.

You can select a range of applications to display. Setting Show Only Active Licenses to Yes displays the current license code for an application. Setting this field to No displays information on current and expired license codes for applications. Records display in descending order of the registration date. If there are multiple records for one application, the record with the latest registration date displays first.

The report includes the application description and version, license code and type, number of licensed users, registration and expiration date, user ID of the person who registered the application, audit date information, and any changes to license information.

Fig. 7.10
Licensed Application Report (36.16.10.8)



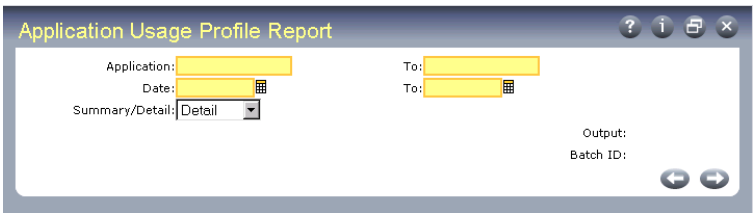
Application Usage Profile Report

After you install and register an application, the software keeps statistics about your application use. The statistics include:

- Licensed application name
- Menu item executable program name
- Number of times the menu item is accessed
- Percentage of the application in use at the time of reporting

You can use Application Usage Profile Report (36.16.10.8) to display the recorded information for each licensed application.

Fig. 7.11
Application Usage Profile Report (36.16.10.8)



You can generate the report in summary or detail format. Summary reports display only the module, access count, and percentage of application use. Detail reports display all recorded information about application usage.

Detailed License Violation Report

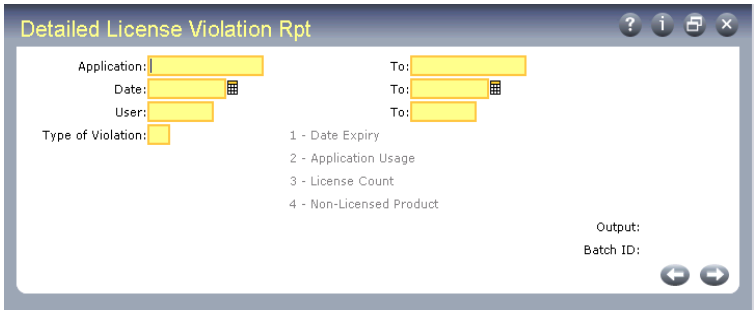
Use Detailed License Violation Report (36.16.10.13) to display information about license violations, including:

- Violation date, time, and error message
- User ID and name of the person who is in violation
- Violation type (for example, application usage or license count exceeded). See “Violation Types” on page 86.
- The total number of sessions and users logged in at the time of violation
- Session ID at the time of violation

- Percentage of the application in use at the time of violation

Detailed license violation reports let you select a range of registered applications registered, dates, user IDs, or violation types on which to report.

Fig. 7.12
Detailed License Violation Report (36.16.10.13)



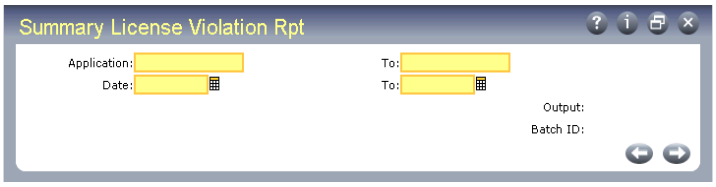
Summary License Violation Report

Use Summary License Violation Report (36.16.10.14) to display:

- Application name, version, and license type
- Violation date
- Total number of violations
- Total number of violations by violation type
- Maximum number of licensed users logged on during a period or the *high water mark*
- Total number of licensed users

Summary license violation reports let you specify the application and the period you want the report to cover.

Fig. 7.13
Summary License Violation Report (36.16.10.14)



If you do not specify an application, all violations for all applications display. If you specify an application, but no dates, all violations for that application display.

If you run either report and there are no violations to report, the following message displays:

```
No violation observed.
```

Audit Reporting

The system provides programs for QAD auditors to use when the auditors gather statistical information on customer use. The programs are not accessible to users. The statistical information is for QAD auditing purposes only.

User Monitor Inquiry

User Monitor Inquiry (36.16.12) displays users currently logged in, along with the:

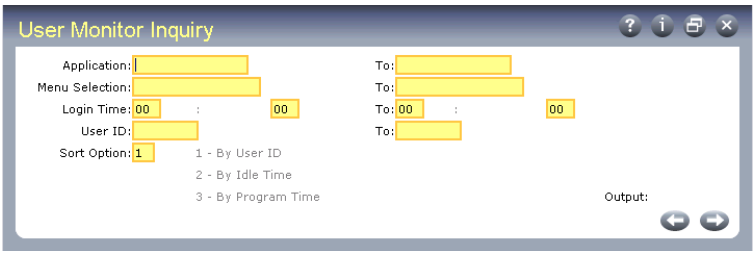
- License type and count for the application
- Program names and menu numbers they are currently executing
- Session ID and user interface type for the session
- Time since they started the current program or menu
- Amount of time they have been idle if no program is selected

Note This inquiry represents a single point in time, not a continuous system record or audit trail.

By monitoring user and program activity, the system administrator can identify users in violation of license agreements and minimize unnecessary overhead during peak system usage.

You can enter a combination of log-in time and users, applications, or menu selections to view details of a specific log-in scenario.

Fig. 7.14
User Monitor Inquiry (36.16.12)



Application. Enter the application name for which you want information to display. You can enter a range of applications by specifying the first application to display in this field and the last application to display in the To field.

Menu Selection. Enter the menu selection for which you want details to display. Leave blank to begin with the first menu matching the other selection criteria.

Login Time. Enter the log-in time for which you want details to display.

Enter the time based on a 24-hour clock in HH:MM format. For example, enter 1:30 pm as 13:30.

User ID. Enter the ID of the user for whom you want details to display. Leave blank to begin with the first user ID matching the other selection criteria.

Sort Option. Enter the number that corresponds to the way you want to arrange information in the User Monitor Inquiry. You can sort by:

- User ID, which sorts the data in alphabetical order by user ID.
- Idle Time, which sorts the data by the length of time a user has remained on a menu. The user with the longest idle time displays first.
- Program time, which sorts the data by the length of time a user has remained in a program. The user with the longest program time displays first.

Setting Up Multiple Time Zones

Accommodating variations in local time is a special global business challenge. The Multiple Time Zones Setup menu (36.16.22) lets you create and maintain time zone data.

- Use Multiple Time Zones Maintenance (36.16.22.1) to define and maintain multiple time zones, including the changes required by daylight savings time.
- Use Multiple Time Zones Inquiry (36.16.22.2) and Multiple Time Zones Report (36.16.22.3) to display and report time zone information.
- Use Multiple Time Zones Load Utility (36.16.22.13) to load sample time zone data.
- Use Database Control (36.24) to specify a server time zone for the database. See “Defining Database Control Settings” on page 97 for details.

Important You should restrict access to these programs, with the possible exception of the report and inquiry. Do not change time zone information without carefully evaluating the impact.

The optional Service/Support Management (SSM) module provides additional functionality related to time zones. If you are using SSM, you can activate the Multiple Time Zone (MTZ) option in Service Management Control (11.24). When MTZ is active in SSM, time zones can be associated with customers, end users, and service engineers and affect the processing of service calls.

See *User Guide: Service/Support Management*.

When MTZ is activated through SSM, the server time zone is set in both Database Control and Service Management Control. If you try to change the server time zone when it is set from SSM, an error message displays.

Multiple Time Zones Maintenance

Use Multiple Time Zones Maintenance (36.16.22.1) to define and modify time zones.

Note The Multiple Time Zones Load Utility creates sample data upon which you can base your own time zones.

This program supports two ways of setting up a time zone:

- In the simplest format, you can base a time zone on an offset from GMT.
- The system can also track daylight savings time adjustments from a baseline you set.

If you choose the second approach, you must specify when the change in time occurs. You can also use effective dates with time zone information, if the start and end points for daylight savings time only apply for a range of years.

After you define the time zones, you can generate reports with Multiple Time Zones Report (36.16.22.3).

Fig. 7.15
Multiple Time Zones Maintenance (36.16.22.1)

Time Zone. Enter an eight-character label identifying a time zone.

Description. Enter up to 40 characters describing this time zone. The description appears in the time zone lookup.

Auto Period Adjust. This field indicates whether the system should adjust the time zone you are defining for a given period—usually daylight savings time or its equivalent.

Yes: Define the period to be adjusted in the subsequent detail frame.

No: Time Period defaults to STD (standard). You cannot change it.

Time Period. This field is editable if Auto Period Adjust is Yes. Valid choices are STD for standard time, Day for daylight-saving time, and Sum for summer time. You can define details for two periods: a standard period, and a special adjusted period for daylight savings or its equivalent. This field determines which of the detail fields are required.

Note Set up values for time period as language details to reflect the terms you use.

Start Year. Enter the beginning year of the range associated with this time zone definition. In some countries, the implementation of time zones varies from year to year. Using start and end dates, you can set up multiple records effective at different periods of time.

End Year. Enter the ending year of the range associated with this time zone definition. If you do not know when the current definition ceases to be effective, use an end year such as 9999.

GMT Offset. Enter the actual offset in hours and minutes from Greenwich mean time (GMT) for this time zone. Enter this number with either a plus sign (+) or minus sign (–) indicating the direction of the offset.

GMT is the base for establishing the relationships among time zones and is never affected by daylight-saving time adjustments.

Start Period. When Auto Period Adjust is Yes, enter the first day of the week when the change of time occurs in MM/DD format. For the United States, daylight-saving time normally begins on the first Sunday in April—identified by a start date of 04/01—and ends on the last Sunday in October—identified by a start date of 10/25.

Note Use the MM/DD format regardless of the date format you use.

This field, in conjunction with the Weekday and Time fields, identifies precisely when the time change occurs.

Note In the U.S., time changes always occur on Sunday (1).

Weekday. When Auto Period Adjust is Yes, enter a number from 0 to 7 indicating the day of the week—identified by the Start Period field—when the time change occurs.

- Enter 0 if the change occurs on the date in the Start Date field, regardless of the day of the week on which it falls.
- Enter a number in the range 1-7 corresponding to Sunday through Saturday if the change occurs on a certain day of the week.

Time. When Auto Period Adjust is Yes, enter the exact time of day—identified by the Start Period and Weekday fields—using a 24-hour clock, when the time change occurs. Enter this time in standard time.

In the United States, enter 02:00 when switching from standard time to daylight-saving time, but 01:00 when switching from daylight savings time back to standard.

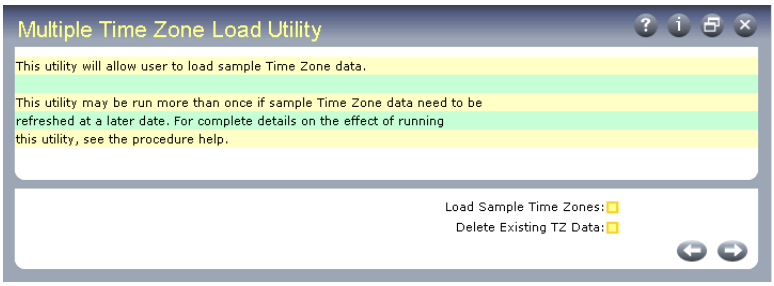
Multiple Time Zone Load Utility

Use the MTZ Load Utility (36.16.22.13) to load a set of sample data based on a snapshot of time zone information. The data assists in the setup process and is a sample only.

After you load this data, verify that the time zones are valid and appropriate for your business practices. Use Multiple Time Zones Report (36.16.22.2) or Inquiry (36.16.22.3) to review definitions and ensure they conform to your requirements. Each organization is responsible for maintaining and updating time zone data to correspond to changing realities and business requirements.

If needed, you can delete existing time zone data and reload the sample data.

Fig. 7.16
Multiple Time Zone Load Utility (36.16.22.13)



Load Sample Time Zones. Yes indicates you want the system to load sample time zone data. You can use this data as the basis for your own time zone maintenance.

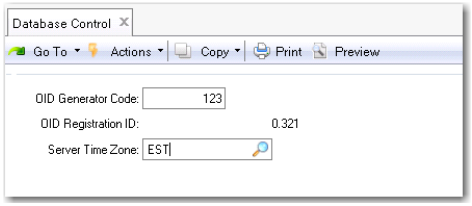
After loading, verify that the time zones are valid and appropriate for your business. Use Multiple Time Zone Report or Inquiry to ensure the definitions conform to your requirements.

Delete Existing TZ Data. The system checks this field only when Load Sample Time Zones is Yes. If you are loading time zone data, you can also delete current time zone definitions. Use this feature if you want to reinitialize the sample data.

Defining Database Control Settings

Use Database Control (36.24) to set the time zone of the database server and to register a code for defining unique record identifiers. You should do this before defining users since the time zone specified here defaults when new user records are created.

Fig. 7.17
Database Control (36.24)



OID Generator Code. The OID Generator Code in Database Control is used to assign unique object identifiers (OIDs) to database records for auditing purposes. The code is assigned during system implementation.

Based on the OID generator code, the OID fields in the database are populated using an algorithm that ensures uniqueness across all records, tables, and QAD ERP application databases within the company. The value stored in the OID field for each record has the following decimal format:

<date><seq_value>.<registration_id>

Where:

<date> is the server date with format yyymmdd.

<seq_value> is obtained from a Progress database sequence.

<registration_id> identifies the origin of the OID value.

The registration ID is derived from the OID generator code by reversing the digits of the generator code value and placing the decimal point in front of the result.

Server Time Zone. Enter the time zone associated with the server machine for the current database. The system verifies that this is a valid time zone defined in Multiple Time Zones Maintenance (36.16.22.1).

If you are using the optional Service/Support Management module and the Multiple Time Zone option is activated in Service Management Control (11.24) for any domain in the database, this field cannot be modified here. Instead, you must use the Server Time Zone Change Utility (11.21.22.22).

See *User Guide: Service/Support Management*.

When a new user is created in User Maintenance (36.3.1), the user time zone defaults from the server time zone.

See “Defining Users” on page 155.

Chapter 8

Reports and Utilities

This chapter includes information on master table audit reports, delete/archive utilities, and operating system commands.

Generating Master Data Reports

Use the Master Data Reports (36.17) menu to generate audit trail reports showing modifications to master tables, as well as reports showing master comments and control program settings.

Auditing Reports

Use audit trails to track and log which users have made changes to fields in master tables. The system tracks high-level information for changes to all master tables.

To maintain detailed information for a critical subset of master tables, set Audit Trail to Yes in Domain/Account Control (36.1). Table 8.1 lists the database tables that the system tracks:

Table 8.1
Audited Tables

Table	Description	Table	Description
ad_mstr	Addresses	plsd_det	Product Line Sales
bk_mstr	Banks	plt_det	Product Line Taxes
bom_mstr	Bills of Material	pl_mstr	Product Lines
cm_mstr	Customers	ps_mstr	Product Structures
cp_mstr	Customer Items	ptp_det	Item Planning
ct_mstr	Credit Terms	ptr_det	Item Routings
cu_mstr	Currency	pts_det	Item Substitutes
dpt_mstr	Departments	ptt_det	Item Taxes
ea_mstr	Earnings	pt_mstr	Items
ecm_mstr	Engineering Changes	slr_mstr	Site Linking Rules
emp_mstr	Employees	slrd_det	Site Linking Details
eu_mstr	End Users	spd_det	Salesperson Commissions
exr_rate	Exchange Rates	spt_det	Cost Simulation Items
is_mstr	Inventory Status	tax_mstr	Taxes
mu_mstr	Monetary Union	vd_mstr	Suppliers
pac_mstr	Purchase Approval Codes	vp_mstr	Supplier Items
pc_mstr	Price Lists	wc_mstr	Work Centers
pld_det	Product Lines		

The audit record contains the user ID, table name, field name, and old and new data values.

Review modifications to tracked master tables with either of the following:

- Use Master Data Audit Report (36.17.1) to print changed records in master tables. The report includes the database table name, current version of the changed record, user ID of the person who made the change, and date.
- Use Master Table Audit Detail Report (36.17.2) to show details about audited changes when Audit Trail is Yes in Domain/Account Control. The report includes current and previous versions of the record, with the time and date of any changes.

The system offers other auditing functions:

- Auditing information for unposted general ledger (GL) transactions is maintained when GL Transaction Audit Trail is Yes in the General Ledger Control (25.24). This data also displays on the Master Data Audit Detail Report.
- Use Change Tracking Maintenance (36.2.22) to track changes to sales order detail fields. See “Tracking Changes” on page 38.

Other Reports

Use Master Comments Report (36.17.5) to print the text of master comments selected by a range of references and by type and language.

Use Control Tables Report (36.17.6) to generate a report listing the current values defined for all control tables in the system. This report is especially important during implementation. It enables you to verify that settings are appropriate for your business environment.

Using Delete/Archive Utilities

Audit Detail Delete/Archive

To delete data from an audit log, use Audit Detail Delete/Archive (36.23.1). This program works differently from other delete/archive functions. It does not delete each record specified. Instead, for each unique combination of user ID, table, and field, it keeps the latest record and deletes/archives the rest.

Use this function to produce a report of records before deleting them.

See “Audit Detail Delete/Archive” on page 78 for an exact procedure.

GL Transaction Delete/Archive

All general ledger transactions are stored in the unposted transaction table until they are posted. Review unposted transactions using Unposted Transaction Inquiry (25.13.13).

To review or delete/archive transactions created in modules other than the general ledger, use GL Transaction Delete/Archive (36.23.2). Use this program when:

1. You are not using the General Ledger module to delete GL transactions created in other modules.
2. You implemented other modules prior to implementing the General Ledger. Before implementing General Ledger, delete the GL transactions in the unposted transaction table. These are reflected in the beginning balances you enter.

Using Operating System Commands

The Operating System Commands menu provides four ways to access the operating system and execute commands directly from the QAD application. Use them as a convenient way of viewing and manipulating information.

- Use Exit to Operating System (36.22.1) to invoke a UNIX or NT session. To return to the QAD application, enter Exit.
- Use Program Execute (36.22.3) to run a Progress program. If the program is not in the current directory, specify the path.
- Use Program/Text File Display (36.22.4) to display the content of an ASCII file, such as a program or print file. If the file is not in the current directory, specify the path.

  Note Add this function to the User Menu so that users can generate reports to a file and quickly review the content.
- Use Disk Space Inquiry (36.22.13) to execute an operating system command to display statistics regarding the current database file size.

Chapter 9

System Cross-Reference

The System Cross-Reference function lets you identify how and where fields and tables are used.

Using System Cross-References

The System Cross-Reference menu (36.18) contains programs that identify how and where fields and tables are used within the system.

System cross-reference activities can be customized to reflect your system setup. This lets you update cross-references when you add or change menu items. If you do not customize the system, you can use the cross-reference as it is.

The cross-reference database requires about 50 MB of disk space, and consists of a set of reports summarizing database relationships such as:

- Which X and Y are used by this Z? X, Y, and Z can be tables, fields, menu items, or programs. *Used* can mean referenced, updated, or called.
- Which database tables are referenced or updated by this menu item?
- Which menu items call this field?
- Which program source files use this include file?

You construct a cross-reference in two steps:

1. Compile the entire system.
2. Build a bill of material from the menu structure.

The end result is a bill of material for each program, in which all programs called by the initial program are components, as well as fields called or updated by those programs.

Cross-reference reports provide different ways of organizing the bill of material.

Background

The system consists of approximately 6200 programs that call some 10,000 fields. The programs consist of normal, executable Progress programs (`.p` files) and include files (`.i` files), which can be called from many different `.p` files.

The menu system calls approximately 1400 of those 6200 programs. These called programs call numerous other `.p` and `.i` files. Progress programs can be nested, enabling you to place `.i` files within `.i` files, and so on.

These Progress programs read or change information in database tables, such as the item master (pt_mstr) or the printer master (pr_mstr). The database tables consist of records containing entries in a group of fields.

When Progress is compiled, the list of programs called and the tables and fields read or potentially updated by those programs can be output. This output, along with QAD-supplied utility information, is the source of the cross-reference.

Table, Field, and Menu Reports

The eight cross-reference reports answer such questions as “What does this table, field, message, menu item, or program do?” The syntax is XYZ. For example, the Tables/Fields by Menu Report tells you what tables X and/or fields Y are called by or updated by menu item Z. Similarly, Menu Item by Message Report tells you which menu items X/Y call a particular message Z.

Table 9.1
Table, Field, and Menu Reports

Program Name	Description
Tables/Fields by Menu Report (36.18.1)	Shows what tables or fields are referenced or updated by programs called by a top-level menu. Limit searches further by execution file, database table, and field. Report includes the type of actions performed by the selected programs on each table or field listed. Action types are create, search, update, delete, and access.
Tables/Fields by Program Report (36.18.2)	Similar to 36.18.1, but not limited to menu-level programs. Shows what tables or fields are referenced or updated by the named Progress program. Before running this report for a top-level program, first use Program Source File Report (36.18.16) to generate a list of subprograms called by the program. Then, run Tables/Fields by Program Report for each relevant subprogram.
Menu Items by Field Report (36.18.4)	Shows which menu items call a field or range of fields. Further limit searches by execution file and database table. Shows field name and table, calling menu item, and kind of action performed. Action types are create, search, update, delete, and access.
Menu Items by Table Report (36.18.5)	Similar to 36.18.4, but limited to a database table or range of tables, rather than fields. Shows which menu items call a table or range of tables. Further limit searches by execution file and menu item. Shows table name, menu item, execution file, and kind of action performed. Action types are create, search, update, delete, and access.
Menu Items by Message Report (36.18.6)	Shows which menu items call a particular message or range of messages. Further limit searches by menu and execution file. Shows message numbers and message text.
Messages by Menu Item Report (36.18.8)	Shows all the messages called by a particular menu item. Further limit searches by menu and execution file. Shows message numbers and message text.

For all reports, the top-level selection is the first one searched. To speed up processing, enter values in the top level.

Using Program Reports

Program reports list all programs—`.i` files and `.p` files—called by a menu item.

Table 9.2
Program Reports

Program Name	Description
Programs by Field Report (36.18.13)	Shows all programs that call a particular field or range of fields. Further limit searches by table name and program name. The report includes the following: • The name of the database table to which each selected field belongs. • The names of the programs and subprograms that reference each selected field. • The types of actions performed on selected fields by each program or subprogram listed. Action types are create, search, update, delete, and access. This program may be useful when a field characteristic has been changed and the programmer wants to know what programs are affected. When you generate a report on programs that reference a specific field such as pt_part, programs using phrases like `where so_part=pt_part` are not included in the report.
Programs by Table Report (36.18.14)	Similar to 36.18.13. Shows all programs that call a particular database table or range of tables. Further limit searches by program name. Useful when a table has changed.
Program Source File Report (36.18.16)	Creates a list of program components, or bill of material, for a specified program or range of programs. Shows all component parts, including nested executable files and include (`.i`) files, that are directly called by the specified programs.
Program Run Report (36.18.17)	Creates a multilevel list of components, or bill of material, for a specified program or range of programs. Shows all component parts, including nested executable files and include (`.i`) files, that are either: • Directly called by the specified parent program • Indirectly called by subprograms or include files that are, in turn, called by the specified parent program Use the Levels field to specify the number of levels to include in the report. For example, set Levels to 1 to list only the subprograms and include files directly called from the parent program.
Source File Where-Used Summary (36.18.19)	Shows which executable files use a specified source (`.p`) or include (`.i`) file or range of files. Useful if you change an include file and want to see the executable files affected. This program does not list intermediate include files. Use Source File Where-Used Detail (36.18.20) to generate a report on intermediate include files as well as top-level program files.
Source File Where-Used Detail (36.18.20)	Similar to 36.18.19. Shows which executable files use a specified source or include file; also shows intermediate include files. Use the Levels field to specify the number of levels to include in the report. For example, set Levels to 1 to list only the executable files that directly call the specified source or include files.

Program Name	Description
Run Program Where-Used Detail (36.18.21)	Shows which source (`.p`) and include files (`.i`) reference a specified subprogram. Lists both top-level source files and intermediate include files. Useful if a called program has changed, and you want to check the behavior of the calling programs. Use the Levels field to specify the number of levels to include in the report. For example, set Levels to 1 to list only the files that directly call the specified subprograms.
Program Summary Bill File Create (36.18.23)	Creates a list of components, or bill of material, for a specified program or subprogram, showing all files in the order in which they are called. List includes all subprograms called by the specified parent program, as well as fields updated by those subprograms. Can include multiple calls of the same file. Report output is placed in an ASCII file, where you can manage it using operating system tools. For example, if you change the name of a called program, use Program Summary Bill File Create to make sure you change each instance of it in the source code.

Updating the Cross-Reference

The cross-reference is built by compiling programs, then checking the compiled programs against the menu. If you change menus or change programs, rebuild the cross-reference using Cross-Reference Update Menu (36.18.24).

Rebuild cross-references as follows:

1. If the source has changed, run Cross-Reference Update from Source (36.18.24.1).
2. Run Missing Component Program (36.18.24.15), Missing Menu Execution File (36.18.24.16), and Programs with No Menu (36.18.24.18) reports.

   These reports show any errors in menu or program listings. Missing Menu Execution File Report, for instance, shows names of programs called by the menu that do not exist.
3. After making corrections, add parent-component relationships not included in step 1. Missing parent-components are supplied by the cross-reference.
4. Run Menu Item Cross-Reference Create (36.18.24.3) to link all cross-reference items with the menu.
5. Delete obsolete cross-reference items.

Chapter 10

Application Server

This chapter includes information on setting up application server definitions used with the Progress AppServer.

Progress AppServer

The Progress Open Application Server, or AppServer, is a brokered collection of 4GL engines that can execute Progress programs on the server in response to remote client requests. Each AppServer instance is identified by a unique name, and contains a broker that manages a pool of 4GL engine processes, each of which is available for processing client requests.

The client connects to an AppServer indirectly through the Progress Name Server. This provides for location transparency (and also provides the logical basis for load balancing and failover) since the clients do not need to know the host and port of the AppServer broker. The client only needs to know the unique name of the AppServer broker, which is used by the Name Server to determine the broker's host and port.

Each AppServer instance can be configured to have its own set of parameter values, such as the PROPATH, database connections, startup/shutdown procedures, and log files. These parameter values are specified in the `ubroker.properties` file, located in the `DLC\properties` directory, where `DLC` is the Progress installation directory.

See the Progress documentation for more information on setting up and using AppServers.

One extremely useful example of the AppServer is to improve the throughput speed of the processing-intensive task of running material requirements planning (MRP). The AppServer can distribute processing load across multiple threads, dramatically improving performance.

See *User Guide: Manufacturing* for information on MRP.

As an example of how an AppServer can be used, this chapter includes instructions for setting up an AppServer to support enhanced MRP performance.

See "Example: Using an AppServer to Run MRP" on page 111.

Before you can run applications using a Progress AppServer, the AppServer instance must be defined in AppServer Service Maintenance (36.19.1).

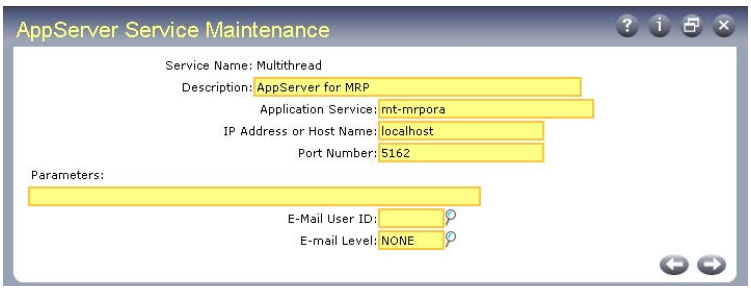
Defining the AppServer

Use AppServer Service Maintenance (36.19.1) to define the information needed for the system to connect to a Progress Application Server.

You can specify a set of standard connection parameters used to connect to this server. Optionally, you can also define server-specific parameters required by the AppServer.

Note The example shown in Figure 10.1 includes the data you would enter to define an AppServer used to improve MRP performance. See page 111.

Fig. 10.1
AppServer Service Maintenance (36.19.1)



Service Name. Enter a name to identify this application server.

Description. Optionally enter a description of the application server.

Application Service. Enter the name of the Application Server defined in the `ubroker.properties` file during configuration of the AppServer.

IP Address or Host Name. Enter the IP address or host name used as the `-H` parameter when connecting to this application server. This is the IP address or host name of the machine on which the application server is running. If the AppServer is running on the same machine as QAD Enterprise Applications, enter localhost.

Port Number. Enter the port number used when connecting to this application server.

- If you are running a Progress name server, enter the name server port number. The default value is 5162.
- Otherwise, enter the port number on which the AppServer is running.

Parameters. Optionally enter any other parameters required when connecting to this application server.

E-mail User ID and E-mail Level. These fields are not implemented and have no effect on processing.

Example: Using an AppServer to Run MRP

This section shows a practical example of how to set up an AppServer to dramatically improve the performance of MRP.

To use the MRP AppServer, you need to perform three main tasks:

- Modify the `ubroker.properties` file for the AppServer instance.
- Configure the AppServer.
- Start and stop the AppServer as required.

See *User Guide: Manufacturing*.

Modify the Properties File

To set up an AppServer to support MRP processing, you must add a set of parameters to the Progress `ubroker.properties` file to identify information about the AppServer instance.

You can modify `ubroker.properties` in two ways:

- Manually edit the file.
- Use the Progress Explorer tool to change parameters through a graphical user interface.

  Note The Explorer tool is available only on Windows.

Progress Explorer can also be used to start and stop the AppServer, and for remote administration.

1. Choose Start|Programs|Progress|Progress Explorer Tool.
2. Choose File|Connect.
3. Specify the host name and Admin Server port of the machine you want to administer remotely.
4. Enter a valid user ID for the remote machine and a password, if required.

Configuring the AppServer

Improved MRP performance requires a single AppServer with multiple threads, which is used to execute the programs that process planning data when you run MRP. Use the following instructions to configure that AppServer.

All Installations

Use this procedure to configure an AppServer instance for all QAD installations. If you have an Oracle installation, additional configuration tasks are required.

In the Progress example shown below, the name for the AppServer instance is `mt-mrppro`. However, you can use any name, as long as all references to the name are consistent.

Add an entry for the required AppServer instance to the `ubroker.properties` file in the `DLC\properties` directory. You can copy the following text into the file. Be sure to change the parameters to match your environment.

See “Additional Oracle Tasks” on page 114.

Note Parameter changes are described after the sample text.

Note Separate examples are provided for Progress and Oracle environments.

Progress Example

```
[UBroker.AS.mt-mrppro]
  appserviceNameList=mt-mrppro
  brokerLogFile=$WRKDIR/mt-mrppro.broker.log
  controllingNameServer=NS1
  initialSrvrInstance=12
  maxSrvrInstance=20
  minSrvrInstance=12
  portNumber=50000
  PROPATH=/dr05/mfgpro/pro/eb2:/dr05/mfgpro/pro/eb2/us/bbi:
    ${PROPATH}${WRKDIR}
  srvrConnectProc=pxldgbl.p
  srvrLogFile=$WRKDIR/mt-mrppro.server.log
  srvrMaxPort=50202
  srvrMinPort=50002
  srvrStartupParam=-c 30 -znotrim -d mdy -yy 1920 -Bt 350 -D 100 -mmax 3000
    -nb 200 -s 63 -noshvarfix -pf /dr05/mfgpro/eb2/Production.pf
  uuid=fdf73fbf039907:6ce891fc:ec7f530e95:-7eed
```

Oracle Example

```
[Environment.mt-mrpora]
  ORACLE_BASE=/dr02/apps/oracle/
  ORACLE_HOME=/dr02/apps/oracle/8.1.7
  ORACLE_SID=mrp
  NLS_LANG=AMERICAN_AMERICA.WE8ISO8859P1
  NLS_NUMERIC_CHARACTERS=.,

[UBroker.AS.mt-mrpora]
  appserviceNameList=mt-mrpora
  brokerLogFile=$WRKDIR/mt-mrpora.broker.log
  controllingNameServer=NS1
  environment=mt-mrpora
  initialSrvrInstance=12
  maxSrvrInstance=20
  portNumber=54000
  PROPATH=.:/dr05/mfgpro/eb2:/dr05/mfgpro/eb2/us/bbi:${PROPATH}:${WRKDIR}
  srvrConnectProc=pxldgbl.p
  srvrLogFile=$WRKDIR/mt-mrpora.server.log
  srvrMaxPort=54202
  srvrMinPort=54002
  srvrStartupParam=-Dsrv svub,1 -c 30 -znotrim -d mdy -yy 1920 -Bt 350 -D 100
    -mmax 3000 -nb 200 -s 63 -noshvarfix -pf /dr05/mfgpro/eb2/Production.pf
   uuid=59fdf73fbf039907:6302bfc1:ec513ed2fd:-6fd7
```

The parameters of interest are described below. Parameters not listed should generally not be changed from the values given in the example.

Important The first line of the entry specifies the name of the AppServer instance. If this is changed from the name in the example, be sure to change all other occurrences of this name in the other parameters.

- `BrokerLogFile` and `srvrLogFile` are the two log files for the AppServer instance. They should be appropriately named and located in a convenient directory of your choice.
- `PROPATH` is the Progress path used to locate code to run. This should reference the r-code directory where the QAD software was installed.
- `uuid` is a global unique identifier value associated with this AppServer instance. The Progress tool genuuid should be used to generate a value. This tool can be run from the command line and is found in the Progress `DLC\bin` directory.

  Note If you use the Progress Explorer tool to create the AppServer definition, the `uuid` will be generated automatically.
- `appServiceNameList` should match the AppServer instance name that you have chosen, which is listed in the first line of the properties entry.
- `portNumber` is the port number for the AppServer broker for this instance. Its value can be an arbitrary integer, as long as it does not conflict with any port assignments of other applications running on this machine, including other AppServer instances.
- `srvrMinPort` and `srvrMaxPort` specify a range of port values to use for the 4GL engines spawned by the AppServer instance. The range should be large enough to accommodate the maximum number of 4GL engines that can be spawned—specified by the `maxSrvrInstance` parameter—and should not have any conflicts with ports used by other applications, including other AppServer instances.
- `srvrStartupParam` specifies the Progress startup parameters to be used by each 4GL engine that is spawned. The specific DB, host, and service names should match the values that correspond to your
  QAD Enterprise Applications database installation.

  Other values should remain as specified in the examples.

- `controllingNameServer` specifies the Progress Name Server instance with which the AppServer broker will register its name. The Progress default is NS1.

Since the AppServer broker `mt_mrppro` is used internally by the system, you must use AppServer Service Maintenance (36.19.1) to define an application server connection master record.

See “Defining the AppServer” on page 110.

Additional Oracle Tasks

If you have an Oracle installation, you must perform two additional tasks:

1 Add an Environment entry like the example below to the `ubroker.properties` file:

```
[Environment.MRP_ORACLE]
    ORACLE_HOME=/Oracle/OracleAppServer
    ORACLE_SID=YourSystemIdentifier
    ORACLE_BASE=/Oracle
    NLS_LANG=AMERICAN_AMERICA.WE8ISO8859P1
    NLS_NUMERIC_CHARACTERS=.,
```

Where:

- *`/Oracle/OracleAppServer`* is the directory where the Progress AppServer for Oracle has been installed; for example, `/dr01/app/oracle/product/8.1.7`
- *`YourSystemIdentifier`* is the Oracle system ID (SID) for your system
- *`/Oracle`* is the base Oracle directory, which contains version-specific subdirectories; for example, `/dr01/app/oracle`

See the Progress AppServer documentation.

Note The NLS variables shown are for American English. Be sure to use appropriate values for your language environment.

Starting and Stopping the AppServers

The AppServer instance configured in the example on page 111 can be administered using the `asbman` command (located in `DLC\bin`), which can be invoked from the command line of a DOS window. The `DLC\bin` directory must be in your PATH environment variable in order to run these commands; alternatively, you can change directories to the `DLC\bin` directory to run them. On UNIX, these commands are located in the `DLC/bin` directory, and the user must have Progress administrative privileges to execute them.

Note Click Start|Programs| Command Prompt to launch a DOS window.

Important Make sure that all databases to be connected to the AppServer are running before you start the AppServer.

The command usage is as follows:

- To start an AppServer instance:

  `asbman -i` *`appServerInstanceName`* `-start`

- To stop an AppServer instance:

```
asbman -i appServerInstanceName -stop
```

- To check the status of an AppServer instance:

```
asbman -i appServerInstanceName -query
```

Example To start the agents for the AppServer name used in the sample `ubroker.properties` file shown on page 112, type the command:

```
asbman -i mt-mrppro -start
```

After starting an AppServer, use the `–query` option to check its status, and do not proceed until all of the AppServers are in the available state.

For troubleshooting, verify that the databases that the AppServer connects to are running. Do this by running a Progress client session and trying to connect to the same database servers.

Note For the AppServer instance to run properly, the Progress Name Server must be running. In turn, for the Name Server to run properly, the Progress Admin Server must be running. Although the Name Server and Admin Server are usually configured by default to start up automatically at boot time, it may be necessary to administer them manually. On Windows, these commands are located in the `DLC\bin` directory, and should be run from a DOS window. On UNIX, these commands are located in the `DLC/bin` directory, and the user must have Progress administrative privileges to execute them.

To start, stop, or query the Progress Admin Server, use the appropriate command:

```
proadsv -start
proadsv -stop
proadsv -query
```

Note In a Windows environment, it is recommended that you use Start|Settings|Control Panel| Services to start and stop the Admin Server.

The Progress Name Server will be started automatically during the successful startup of the Admin Server. If it is necessary to start, stop, or query the Progress Name Server (assuming the default NS1 name is used for the Name Server), use the following commands:

```
nsman -i NS1 -start
nsman -i NS1 -stop
nsman -i NS1 -query
```

Chapter 11

User Interface Management

This chapter discusses programs that let you modify the ways users interact with the system through the user interface.

Introduction

The UI: Manager Functions menu provides several programs that let you customize various aspects of the user interface. For example, you can use these programs to design a view, incorporate it into a browse, then attach the new browse to a field.

Table 11.1 lists the user interface manager functions that are described in this chapter.

Table 11.1
UI: Manager Functions Programs

Number	Menu Label	Program
36.20.1	Drill Down/Lookup Maintenance	mgdlfhmt.p
36.20.4	User Tool Maintenance	mgtoolmt.p
36.20.6	Menu Substitution Maintenance	mgmsmt.p
36.20.13	Browse Maintenance	mgbwmt.p
36.20.18	View Maintenance	mgvwmt.p

This menu also contains programs that are not described in this chapter. If you are using QAD Desktop, additional programs support customizing this interface (36.20.10).

See *User Guide: QAD Desktop* for details.

Maintaining Drill Downs and Lookups

Browses display selected data in the form of a table. Two types of browses are available:

- *Look-up browses* return the value you select to the active field in the calling program.
- *Drill-down browses* are more complex. They include more information and can display, filter, graph, or print data.

The field values in the browse can come from a table or a view. A *view* is a table that has selected values from one table or several joined tables.

Use Drill Down/Lookup Maintenance (36.20.1) to assign drill downs or lookups to fields that do not have a browse, to replace a browse, or to delete one.

One of the most common uses of this program is to display generalized codes associated with a field. You can also assign look-ups to any field that acts as an index to a maintenance screen. You may, however, need to write your own custom browse to find and display the data.

See “Adding a Lookup” on page 28.

Most programs attached to a function with Drill Down/Lookup Maintenance display values in a database table. But this is simply a convention. You can attach any Progress function to a field, and this program executes when the user selects Help. For example, you can attach the program `calculat.p` to field pt_avg_int to display a calculator.

Before you can use Drill-Down/Lookup Maintenance, you need to know:

- The name of the field where you want the browse to display.
- The name of the program using the field.

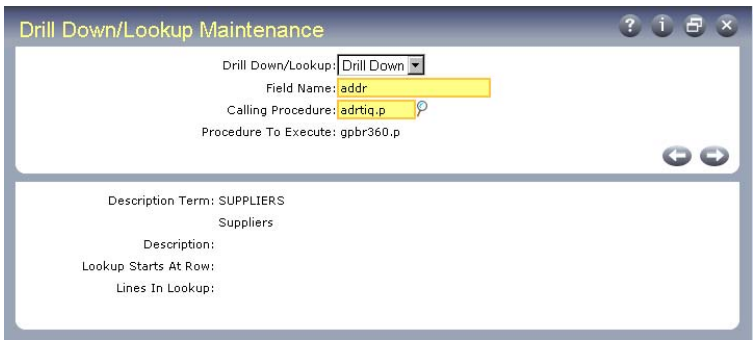
- The program name of the browse to attach. If a lookup is missing for a particular field but exists for a similar one, use Lookup Browse (36.20.3) to determine the program that displays appropriate field values. Then use Drill Down/Lookup Maintenance to specify the same program for the similar field.

See “Creating Browses” on page 124 for details on creating browses.

Determining the name of the program and field depends on the user interface.

- In the Windows and character interfaces, run the program. Press Ctrl+F and note the program and field names that display in the pop-up window.
- In the Desktop interface, run the program. Click on the information button on the right side of the program title bar. The button is identified with the letter i. A screen displays program information, including the program name. To identify the field name, place your cursor over the field where you want to attach the browse. The field name displays.

Fig. 11.1
Drill Down/Lookup Maintenance (36.20.1)



You can assign more than one drill down to the same field. A menu of drill downs appears when you request the drill down. Only one lookup can be attached to a given combination of field and program name.

You can attach browses to fields in any program, including another browse. Drill downs can be nested. A field can call a browse that can call another browse that can call another browse, and so on.

Follow these steps to use Drill Down/Lookup Maintenance to associate a drill down with a field or program:

1. Select Drill Down in the Drill Down/Lookup field.
2. Enter a field name to associate with the browse in Field Name. Leave it blank to associate it with all fields.
3. Enter the program containing the field in Calling Procedure. Leave it blank to attach the browse to all programs using the specified field.
4. Enter the browse name in Procedure to Execute.
5. Optionally, enter a label term in Description Term. The long label contained in this term is displayed as the title in the browse. The default is the browse description term defined in Browse Maintenance. See “Creating Browses” on page 124.

You can access drill downs in four ways:

- Select Drill Down from the Help menu and click on the field.
- Select the Drill Down icon on the toolbar and click on the field.
- Double-click on the field in the browse.
- Select the field and press Alt+F1.

Follow these steps to associate a lookup with a field:

1. Select Lookup in the Drill Down/Lookup field.
2. Enter a field name to associate with the browse in Field Name.
3. Enter the program containing the field in Calling Procedure. Leave it blank to attach the browse to all programs using the specified field.
4. Enter the browse name in Procedure to Execute.
5. Optionally, enter a description for the lookup. This description is for reference only and is not displayed in the lookup.
6. Enter the starting row and the number of lines to display in the browse pop-up window.

Wildcards in Drill Down/Lookup Maintenance

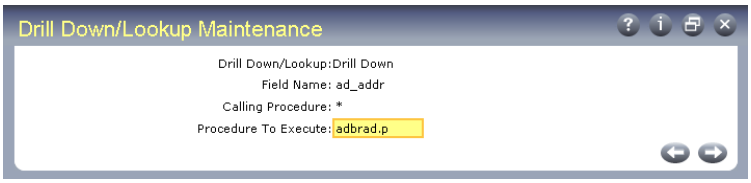
Use wildcards to attach browses to fields in multiple programs. For example, `pp*.p` attaches the drill down to the specified field in all programs starting with `pp` and ending with a `.p` extension.

Possible entries to Drill Down/Lookup Maintenance:

Field	`ad_addr`	`ad_addr`	`ad_addr`
Calling Procedure	`*`	`so*`	`soivmt.p`
Procedure to Execute	`adbrad.p`	`adbrcs.p`	`arbrbl.p`

When you drill down on `ad_addr` in `soivmt.p`, a menu shows all three browses: `adbrad.p`, `adbrcs.p`, `arbrbl.p`. When you drill down on `ad_addr` in a program other than `soivmt.p` but beginning with the letters `so`, a menu shows two browses: `adbrad.p` and `adbrcs.p`. When you drill down on `ad_addr` anywhere else, the browse `adbrad.p` opens.

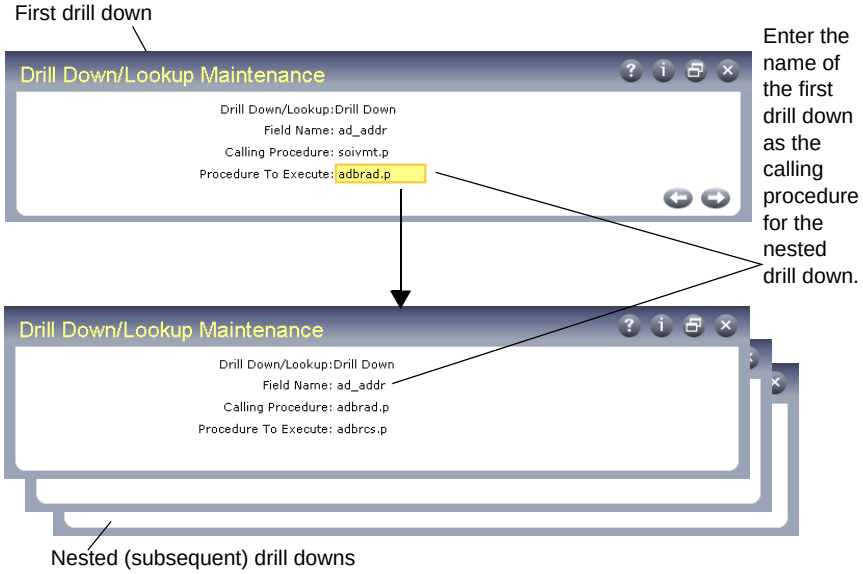
Fig. 11.2
Wildcards in Drill Down/Lookup Maintenance



Drilling Down on Drill Downs

You can nest drill downs. In other words, one drill down can call another, which can call another, and so on. After creating the first drill down, you can assign the others to the same field. Enter the name of the first drill down as the calling procedure for the nested drill down.

Fig. 11.3
Nested Drill Downs



Planning for Upgrades

When you update to a new version, be careful when loading flh_mstr. This table contains the records created by Drill Down/Lookup Maintenance. If you have customized it, make sure that the new version does not overwrite your customization.

Creating Access to Other Programs

User Tool Maintenance (36.20.4) lets you specify programs that can be run from other programs. This makes it easier for you to run frequently used programs.

Note The relationships you define in User Tool Maintenance do not apply to any programs in the character interface and they do not apply to browses.

How you define access to programs and the way you run them varies depending on whether you are using the Windows or Desktop interface.

Windows Interface

In the Windows interface, you can assign up to four buttons and four User Menu items to launch programs of your choice. You assign programs by user and program. You can change buttons for all users or only some. By default, programs assigned to buttons are also assigned to the User Menu.

You can assign images to the buttons to make them easy to identify or use a text label only.

See “User Menu and Function Keys” on page 47.

Warning In the Windows interface, you generally assign browses and inquiries only to toolbar buttons. Running a maintenance program while working in another maintenance program can cause problems and is not recommended.

Desktop Interface

In the Desktop interface, you use User Tool Maintenance to assign links that let you access one program from another. These links display on the bottom of the program screen.

Images do not apply to Desktop. The link displays the text label specified. If no label is specified, the standard menu description from Menu System Maintenance is used.

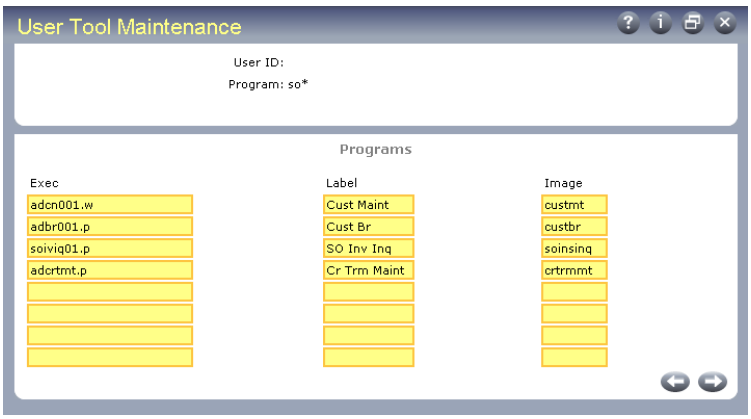
When you click a link, the program opens in a new, detached window. You can run as many detached windows as the system settings allow.

See *User Guide: QAD Desktop* for details on adding links.

User Tool Maintenance

Figure 11.4 illustrates the User Tool Maintenance screen.

Fig. 11.4
User Tool Maintenance (36.20.4)



1. Enter a user ID or leave the field blank to assign the button or link to all users.
2. Enter a program name or leave the field blank to assign to all programs. You can also use wild cards to specify where the options appear. Specifying pp* places the buttons and links in all programs beginning with pp.
3. In the Exec fields, enter the program names (for example, `adbr001`) for the buttons or links to launch.
4. In the Label fields, enter the button or link labels, which you can write as abbreviated program names; for example, Cust Maint.
5. In the Windows interface, optionally enter the bitmap image file names in the Image fields. The image files must be in the user's PROPATH.

Displaying Buttons and Links

You can assign programs to all users (blank user ID) or a specific user. You can also assign programs to a specific program or using wild cards. However, only one set of records displays when a user accesses a program. They system searches for the appropriate buttons or links to display in this order:

1. Specific user ID and specific program name
2. Specific user ID and program name with wildcards
3. Blank user ID and specific program name
4. Blank user ID and program name with wildcards

The system displays buttons or links only for the first available combination it finds. Use User Tool Maintenance in combination with User Function Maintenance (36.4.11) to manage global and local access to programs. Specify the additional programs you want to display in one or the other.

See “User Menu and Function Keys” on page 47.

Setting Up Menu Substitutions

Use Menu Substitution Maintenance (36.20.6) to set up a link between two programs so that when users select one from a menu, the other program displays. This is useful for substituting custom versions of existing programs.

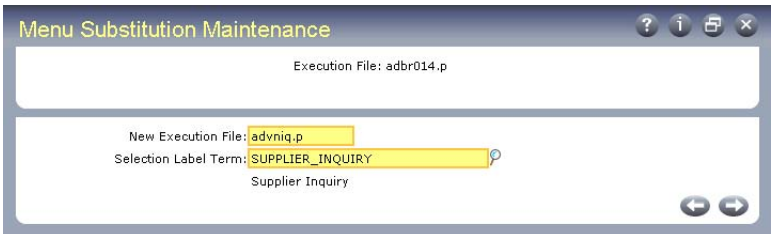
In the Windows and character interfaces, which program is invoked for a particular user depends on whether they have enabled menu substitution. Each user can turn menu substitution on or off in User Maintenance (36.3.1) or, in the Windows environment, from the Options menu.

In the Desktop interface, browses and standard programs are always placed on the menus. Users can find the alternate program using the search function.

Menu substitution affects standard programs in these ways:

- Replaces browses with inquiry programs
- Replaces standard programs with custom versions

Fig. 11.5
Menu Substitution Maintenance (36.20.6)



1. Enter the program name in Execution File. Users selecting this program from a menu will actually be running the one entered into the New Execution File field.
2. Enter the substitute program name in New Execution File. This is the name of the program to replace the one entered in Execution File. Users will run this program when they select the one entered in the Execution File field. You can use wildcards. For example, if you want to replace all inquiry programs with the browse versions, you enter `*iq*` in the Execution File field and `*br*` here.
3. Enter a label term in Selection Label Term. The long label contained in this term appears in the title bar and menu list of the substituted program.

Creating Browses

Use Browse Maintenance (36.20.13) to create browses, which display selected data in the form of a table.

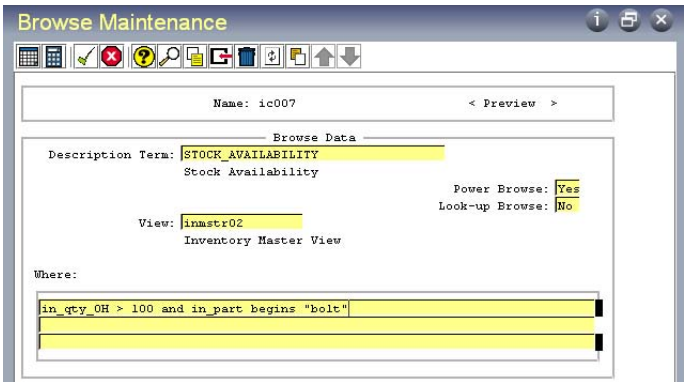
When you create a browse, it is saved in your working directory as a source-code file. The source-code name is the first two characters of the name you entered, then the letters `br` or `lu` (depending on whether you selected power or look up), then any remaining numbers from the name you specified, then the extension `.p`.

Example You create a power browse and name it `ap010`; the system names the code `apbr010.p`. If you selected both power and look-up browses, the system generates two source-code files: `apbr010.p` and `aplu010.p`.

Although you do not need to compile the source code of the browse, you should for better performance. If other users on your network want to use your browse, you must compile it and move it to the network directory. Use the Progress editor to compile the browse.

Note You can access the Progress editor only if your PROPATH is correctly set up to access source files.

Fig. 11.6
Browse Maintenance (36.20.13)



To create or modify a browse:

1 Select or enter a name for the browse. To name the browse, enter two letters and press Enter. The system gives the browse a name that increments by one the number in the file name of the last browse created.

Note Use the existing module mnemonics or make up your own.

2 Press Go. To preview an existing browse, press Enter. Otherwise, press Go again.

Important Previewing a browse can be a time-consuming process because the system generates and displays the browse in runtime.

3 Enter a label term in Description Term. The long label contained in this term is displayed as the title in the browse window.

4 Indicate whether this is a power browse, look-up browse, or both.

5 In View, enter the name of an existing view or a primary table whose data the browse displays. You can see only those views you have access to. If a view exists for a table and the view name is the same as the table name, you have access to only those fields that are available in the view.

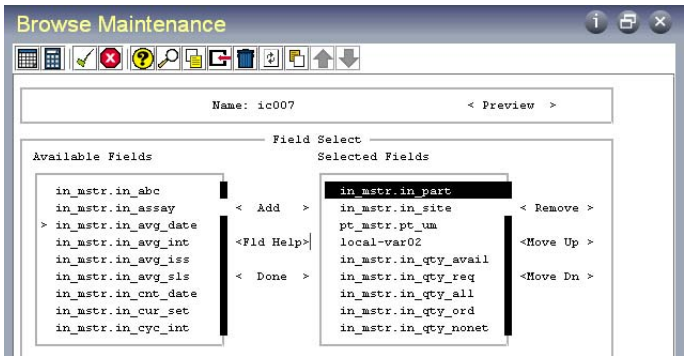
See “Creating Views” on page 127.

Note The view name you enter in View must already be defined in View Maintenance, or you must enter a primary table name.

6 In the Where field, type the selection criteria (optional) to limit the browse’s search to records that meet a certain condition. The criteria in Figure 11.6 would display only inventory balances of bolts greater than 100. Do not put a period (.) at the end of the criteria, because the system adds a `no-lock no-error` statement to the criteria.

7 Press Go.

Fig. 11.7
Browse Maintenance, Field Select



8 Fields from the view or primary table entered in the Browse Data frame display in Available Fields. Include up to 20 fields in your new browse.

- In the Windows interface, select fields to include in your browse by clicking on them and choosing the Add button. To view help on an available field, click on the field and choose the Field Help button.
- In character mode, select a field to include in your browse by using the Up and Down keys to locate it and then press Enter. Multiple fields can be selected. Use the Tab key to choose the Add, Field Help, or Done buttons or to navigate between the Available Fields list and the Selected Fields list.

9 You can use the Move Up and Move Down buttons to arrange the fields in the Selected Fields list. If you want to remove a field from the Selected Fields list, select it and choose the Remove button.

When you have arranged the fields in the order you want, press Go.

10 Enter the column number to take the field values from in Value-Returned Column (optional). The default is the first column of the browse.

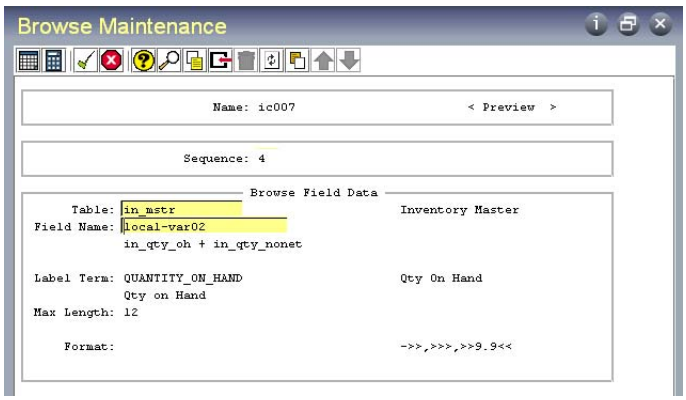
11 In the Sort Columns field, enter the columns you want to have available for sorting. Enter the columns as a comma-delimited list of up to seven numbers. The first field name in the Selected Fields list is column 1, the second is column 2, and so on.

The look-up browse sorts the records on the first column you enter in the Sort Columns field. The remaining columns you enter are listed in the selection list above the browse. Select another column in the list and the browse re-sorts on that column. When it re-sorts, the browse redisplays beginning at the first record. The browse does not redisplay beginning at the record that was selected when the re-sort was initiated. By default, the browse sorts on the first field in the Selected Fields selection list.

Note The Sort Columns field is enabled only for look-up browses.

12 Press Go.

Fig. 11.8
Browse Maintenance, Browse Field Data

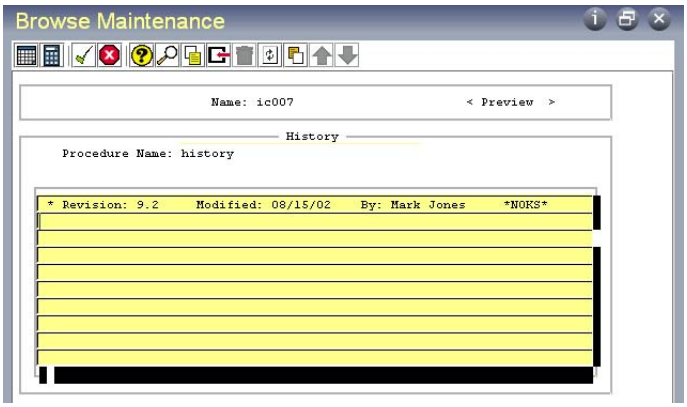


13 Enter a sequence number to access field data.

14 Identify the table and field and change the default field label and format (optional). To control the display length of a label, enter a Max Length value.

15 Press End.

Fig. 11.9
Browse Maintenance, Revision History



16 The program automatically creates a revision history line containing a revision number, the user name (or logon ID), and current date. You can modify this as needed. The revision history is also saved in the source code.

17 Press Go to generate the browse. To save the browse data without generating the browse program press End.

Creating Views

A view is a display of some or all of the fields from one or more tables. You join two or more tables for a view by specifying the relationships between fields in different tables and choosing the type of join to use.

Views are used in browses, which display the fields gathered using views. By choosing which fields to include or exclude in a view, you control which fields are available for a browse to display. By putting security on the view, you can allow users to modify browses, knowing that they can access only those fields that you have authorized.

Use View Maintenance (36.20.18) to create or modify views.

Using Progress Syntax

You use some Progress syntax in creating or modifying views. You must also understand database table and field relationships.

To create or modify a view:

1. Select the table or tables to include in the view.
2. For sequences after the first, specify the type of join to use: inner or outer.
3. Join the tables using Progress logic.
4. Select fields from the tables.
5. Save the view.

Figure 11.10 illustrates how to create a view of selected fields from two tables.

Fig. 11.10
Creating a View by Joining Two Tables

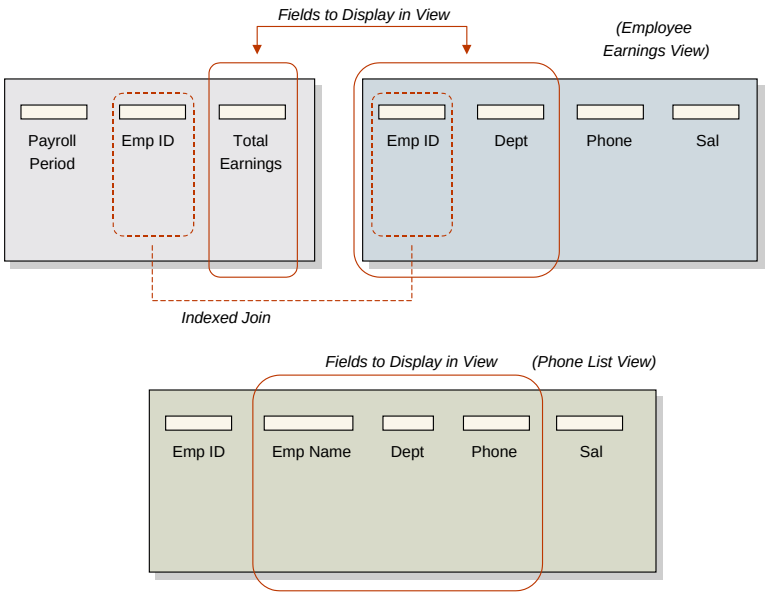
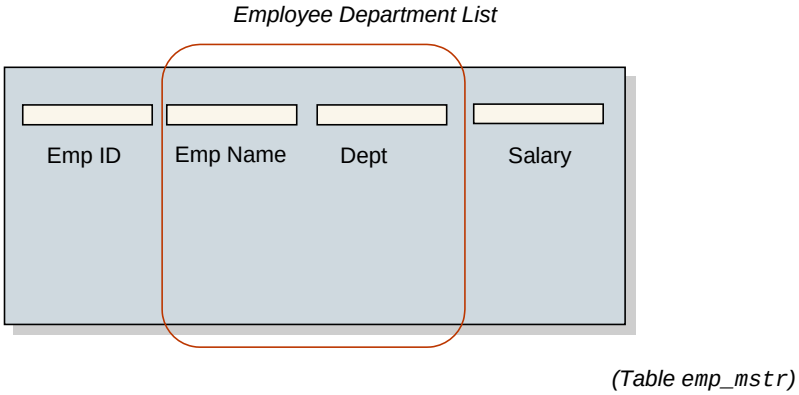


Figure 11.11 illustrates how to create a view of selected fields from one table.

Fig. 11.11
Creating a View from One Table



Using Join Types

When a view includes data from more than one table, you can choose from two types of joins when creating a view:

An inner join returns the records selected for the first table combined with related records selected from the second table. If a record does not exist in the second table, no records are returned. Only related records selected from both sides of the relationship display in the view.

An outer join returns the records found by an inner join. However, in addition, for each value in the first table, it returns unknown values from the second table when no related record is found. As a result, all matching records from the first table are preserved for unmatched records in the second table.

The default join type is inner. Using the outer join can give you more flexibility in displaying information.

Example An inner join between customers and sales orders displays only customers with sales orders. An outer join includes all customers, even those who do not have orders.

Using View Maintenance

Figure 11.12 illustrates View Maintenance (36.20.18).

Fig. 11.12
View Maintenance (36.20.18)

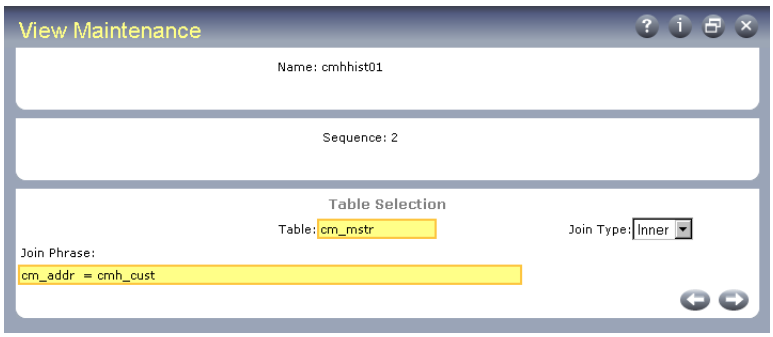
1 Select or enter a view name.

2 Enter a label term in Description Term. The long label contained in this term is displayed as the view label.

3 In User IDs/Groups, enter a user ID to limit user access to the view (optional). You can enter multiple user IDs by separating them with commas.

4 Press Go.

Fig. 11.13
View Maintenance, Table Selection



5 The number you enter in Sequence controls the order in which the table defined in Table is joined to the view.

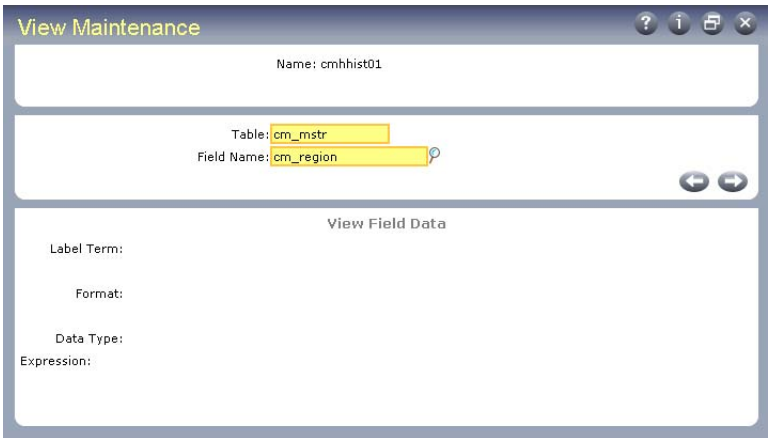
6 Enter a table name.

7 If the sequence is not 1, specify the type of join, either inner or outer. The Join Type field is only enabled when the sequence number is greater than 1.

8 Enter or edit the phrase to join the tables. Use proper Progress syntax. Do not include a Where verb. Join phrases express the field relationships between tables (see Figure 11.10). For a faster display of fields, use indexed fields in the Join Phrase.

9 Press End.

Fig. 11.14
View Maintenance, View Field Data



10 In Field Name, enter a field from one of the tables in the view or enter a local variable. When entering a local variable, name it `local-varnn`, where `nn` is a number incremented by one from the last defined variable.

For example, you see from the look-up browse that the last local variable was `local-var05`; you name your local variable `local-var06`. Use local variables when you want to return a value resulting from an operation on two fields; for example, the quantity required minus the quantity open. Define the operation in Expression.

11 If you entered a local variable in Field Name, enter its Label Term, Format, and Data Type.

Note Search for a label term by entering a portion of a label, then use Next/Previous to display available records.

12 If Field Name is a local variable, you can enter Progress syntax in Expression to define the local variable. Valid expressions include:

- `field1 + field2` (computation, where `field1` and `field2` are fields within the record)
- `>`, `<`, `>=` (operands that perform comparisons)
- Progress functions, such as `substring (field1,1,4)` or `round (field1,1)`

Note Incorrect syntax terminates your session if you attempt to use the view.

Chapter 12

Users and Security

This chapter describes how to set up users and manage different kinds of security.

Security in QAD Enterprise Applications

Security and related technical controls must be viewed within the context of an 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

Availability includes items such as policies and procedures for data, equipment, and infrastructure backup and recovery. Features that can support these items are discussed in other sections of the user guide.

Security properly starts with a comprehensive policy statement that:

- Clearly demonstrates 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 such as those described in this chapter are designed and implemented to support the administrative controls.

This chapter 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 145.

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 is designed to support external requirements. Numerous controls have been introduced 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 Overview

Security options are available on several levels, based on information defined in user master records.

Two types of security are enforced at log-in:

- Log-in security determines whether a user can log in to a session. This level of security is always active and requires that users specify a valid user ID and password before they can log in.

  See “Basic Login Security” on page 137.

  Optionally, system administrators can choose to bypass log-in security and automatically log in valid users based on operating system-level access.

  See “OS-Based Log-in Security” on page 138.

  Note You also should consider additional access security options at the operating-system and Progress levels. See “Operating System and Progress Security” on page 139.
- Domain security limits individual user access to specific domains identified in User Maintenance. Users are limited to authorized domains at log-in. Additionally, the system controls domain access when the user runs Change Current Domain (36.10.13), as well as certain programs that can display or update records from multiple domains.

Other security methods are based on what action the user is attempting within the system. Individual programs let you control access based on individual user IDs and/or user groups:

- Menu security (36.3.10) limits access to menus and menu functions.
- Field security (36.3.19) limits who can update specific fields.
- Entity security (36.3.13) limits who can create GL transactions for a particular entity.
- Site security (36.3.15) limits who can create inventory transactions at secured sites.
- General ledger (GL) account security (36.3.9) restricts access to GL accounts.
- Inventory movement security (36.3.17) lets you grant or deny group members access to shippers and other transactions using specific movement codes at a site.

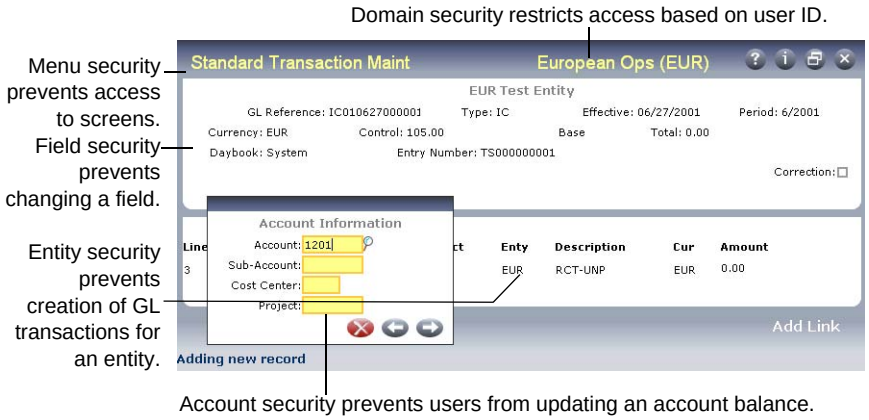
See “Using Security Functions” on page 166.

Note If you use the Sales and Use Tax Interface (SUTI) to communicate tax data with 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* for information on SUTI.

Figure 12.1 illustrates how several different kinds of security can operate at the same time with the same user.

Fig. 12.1
Types of Security



Security functions use user IDs and groups for system elements controlled by menu, site, entity, and so on. The security maintenance function creates a record that pairs a field or function and user IDs or user groups.

See “Using Security Functions” on page 166.

- For menu security, site security, GL account security, and inventory movement code security, specify any combination of user IDs or user group names.
- For entity and field security, specify user IDs.
- For domain security, grant each individual user access in User Maintenance.

See “Controlling Access with User Groups” on page 163.

When a user tries to do something that is controlled by security, the system compares the security records with the ID and groups associated with the current user. If there is a match, the system grants or limits the user’s actions accordingly.

Additional access control features are provide by the QAD .NET and QAD Desktop user interfaces. See the user guides for those interfaces for information.

Important The various security controls are primarily effective within a user session. The 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 application data using other means. See “Operating System and Progress Security” on page 139.

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 required 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

See “Create a Password Strategy” on page 152.

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 log-in 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 log-in. User Password Force Change Utility (36.3.23.12) makes all users or members of specified groups change their passwords.

See “Updating Passwords” on page 160.

Basic Login Security

Typically, a user must enter both a user ID and a password to log in. If the user enters an invalid combination, 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 deactivated, and members of the system administration group are notified by e-mail. The sending address of the e-mail includes the operating system ID of the user who attempted to access the system. Figure 12.2 illustrates how this process occurs during log-in.

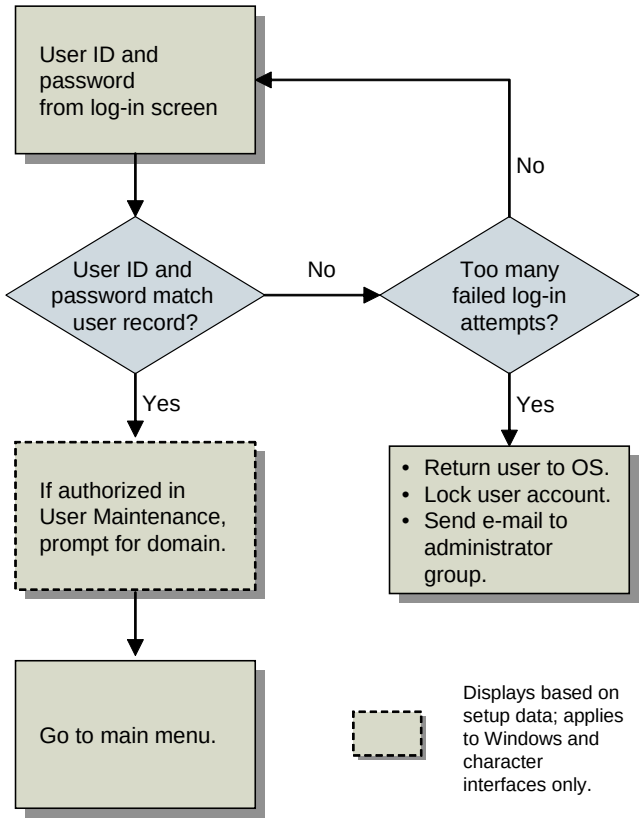
See “Setting Up Security Control” on page 148.

Note To completely or partially bypass log-in security, you can configure the system to allow users to access the system based on operating system user ID. See “OS-Based Log-in Security” on page 138.

Depending on the setting specified in Security Control, the system maintains historical records of successful and failed log-in attempts. Use Logon Attempt Report (36.3.23.1) to view log-in history.

Note In order for the time zone to be properly recorded during log-in and password change, the server time zone must be specified in Database Control (36.24). See “Setting a Default Time Zone” on page 14.

Fig. 12.2
Log-In Validation



This type of log-in security lets you:

- Unless you choose to control access from the operating system level, effectively separate application security from the operating system security. The application user ID does not have to be the same as the user ID referenced by UNIX or Windows. See “OS-Based Log-in Security” on page 138.
- 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 the QAD application presents an additional barrier to an unauthorized user.
- Track unsuccessful log-in attempts to identify possible unauthorized efforts to access the system.

To provide maximum security, the system does not save log-in related data from session to session. User interfaces typically require users to enter both a valid user ID and a password at each log-in unless you choose to control access directly from the operating system level.

OS-Based Log-in Security

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

See “Setting Up Security Control” on page 148.

If they do not use application passwords, this feature essentially allows customers using those interfaces to bypass log-in security completely and rely on operating-system security.

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

When Enforce OS User ID is Yes, the default user ID displayed in the log-in screen is the same ID used by the operating system, and the user cannot change it.

Note This must still be a valid user ID defined in User Maintenance (36.3.1).

Subsequent processing depends on whether a password is specified in User Maintenance or User Password Maintenance (36.3.3):

- If no password is specified in the user record, log-in 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 to blank, you should use caution if Enforce OS User ID is ever changed to No. If you do so without reentering passwords in user records, anyone can gain access by entering just a user ID. When you change the field from Yes to No, the system displays a message to warn you of a potential security compromise. In addition, in Windows environments it is not recommended that you reset user passwords to blank. It is relatively easy to create a new user on an existing Windows machine with an ID that matches one in the system.

Domain Security

Access to domains is controlled at two points:

- During system log-in
- During the session

When your User Maintenance record specifies more than one domain, you can switch domains after log-in using Change Current Domain (36.10.13). However, the system never lets you access a domain that is not authorized in your user record.

See “Changing the Current Domain” on page 13.

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 application-level controls, you should consider additional security at the operating system and Progress levels.

At the operating system level, all 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
- Files used to execute system administration tools such as MFG/UTIL

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 a database directly from Progress

Progress Editor Access

One area of potential security exposure is related to the Progress Editor. By default, legitimate users can access the Progress Editor by exiting from the menu and specifying the appropriate code at the exit prompt. Once a user has accessed the Progress Editor, data can be significantly exposed.

You can use Menu Security Maintenance (36.3.10) to limit access to the Progress Editor just as with standard menu programs:

1. Leave the Menu field blank.
2. Set Selection to 1.
3. Enter user IDs or groups for any users who should have access to the Progress Editor.

See “Assign Access by Menu” on page 168.

Another related control that should be considered is to disallow privileges for users connecting to the 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.

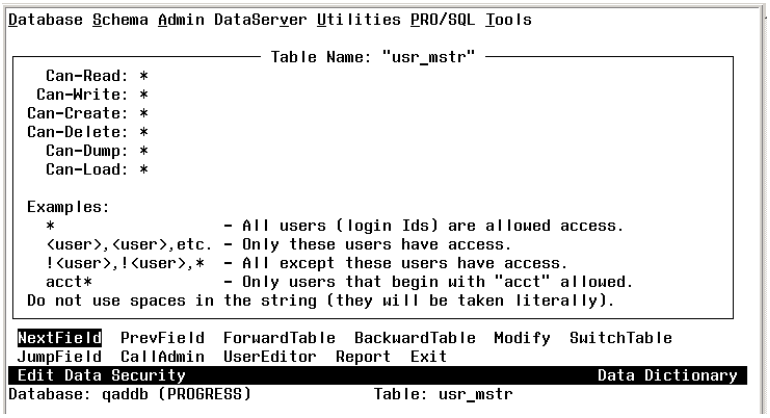
Selecting this option denies all access privileges to the Progress blank User ID by placing a leading exclamation point (!) in each table and field permission specification for the database. See the next section for a more detailed description.

See the “Maintaining Application Security” section in the Progress *Client Deployment Guide* for details.

Progress-Level Database Schema Controls

Progress-level security controls should also be considered for protecting the 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. 12.3
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.

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.

See the Progress *Database Administration Guide* for additional details on these functions.

Progress-Level Database Access

Unless properly controlled, it is possible under certain conditions to start a Progress session and then connect to a database without starting the application. After connecting, there would be no effective controls over accessing private or confidential data, modifying, or deleting records. Since a 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 application 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 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 log-in or the network log-in 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 database, restricting access to the database files, and monitoring the database log file for unusual access events, security exposures from inappropriate access to the database can be substantially reduced.

Workstation-Level Security

Depending on the operating system of the machines that are running the user 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 a 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.

For data integrity reasons—for example, to prevent a user from having a session terminated without saving modified data—this feature applies only when the system is displaying a menu, rather than when a program is executing. To add workstation security for times when a user leaves a computer unattended while a program is running, you can use operating system features.

See “Setting Up Security Control” on page 148.

Windows Systems

In many environments, users run a Windows system; for example, GUI clients, character sessions using a terminal emulator, QAD Desktop sessions using a Web browser, or QAD .NET UI clients. 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 application log-in—to turn off the screen saver.

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

1. Right-click on 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.

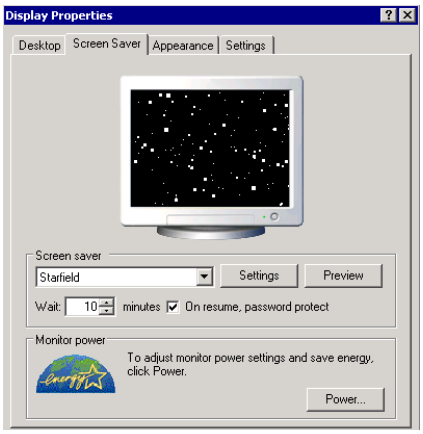
When the screen saver comes on, it can be cleared only when the current user's Windows password is entered, or when an individual with system administrator access overrides the user log-in.

See the Windows online help for more information.

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.

Figure 12.4 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. 12.4
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. See the operating system documentation for your system for information.

Non-Windows Systems

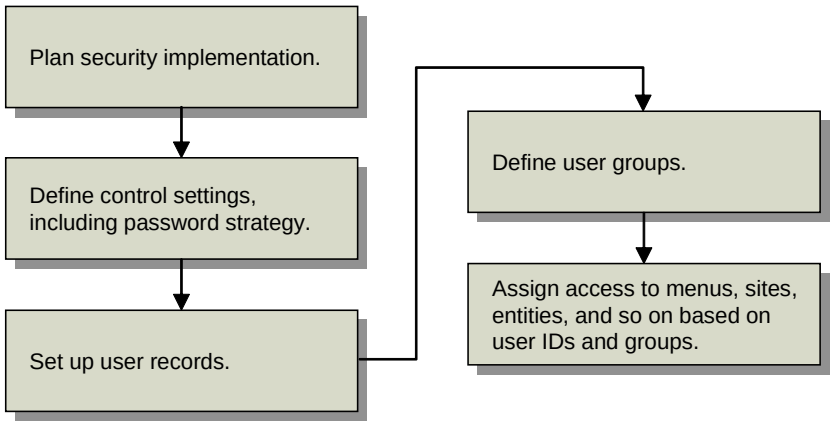
Many standard UNIX workstations—including those provided by HP, Sun, and IBM, which use the Common Desktop Environment (CDE)—offer screen-locking features much like those in Windows. Set up CDE-based machines using the Style Manager icon on the Front Panel. Similar features are also available for some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

See the user documentation for your workstation for specific information.

Security Implementation Summary

Figure 12.5 illustrates a work flow for implementing and using security features.

Fig. 12.5
Security Work Flow



Establish a Security Plan

By default, only log-in security is defined. Once you set up explicit permission for one user to access entities, fields, menus, and so on, all other users are excluded. For this reason, you should have a comprehensive security plan before beginning to set up security records.

The set of checklists provided in this chapter can serve as a starting point for determining the focal points to consider when establishing a plan.

See Table 12.1 on page 146.

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

- Does your company have specific requirements regarding password aging for all its systems?
- Do external regulatory agencies set standards for such things as 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 of the QAD application?

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

For example, user profiles defined in User Maintenance 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.

See “Defining Users” on page 155.

Additionally, be aware that while user IDs and groups are defined for the entire database, group security access is controlled on a domain-by-domain basis. For example, you can restrict a particular group from accessing a GL account in Domain 1—but give the same group access to that account in Domain 2.

See “Streamlining Setup” on page 7.

Implement Your Security Plan

After planning how your security system should operate to meet your company’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 148.
- Set up user records. Depending on your overall security plan, you can define such elements as domain access and group membership, as well as enter temporary passwords for your users. See page 155.

  Note If you want to assign users to groups at the same time you set up user records, you must define groups first. Alternatively, you can just define the users and assign them to groups in User Group Maintenance (36.3.4).
- Based on how you want to control access to functions, define groups using User Group Maintenance. See page 164.
- Use several programs to set up user or group access to menus, fields, sites, entities, GL accounts, and inventory movement codes. See page 166.

Security Planning Checklists

Tables 12.1 through 12.3 summarize the various security controls that should be considered as part of an effective overall information security plan. 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 12.1
Planning, Policies, and Procedures Checklist

Topic	Reference
Review all information security documentation for both QAD 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 QAD-related files to determine the appropriate permission and ownership settings.	"Operating System and Progress Security" on page 139
Document the users who should be permitted access to the application and verify user IDs.	"Defining Users" on page 155
Determine if user groups will be used, and if so document the group names and the user IDs to be assigned to each group.	"Controlling Access with User Groups" on page 163
Consider requirements for policies and/or procedures regarding deactivation of old user accounts. To meet the requirements of many regulated environments, user accounts can be deactivated, but not deleted, once they have been used to access the system.	"Defining Users" on page 155
Define policies and procedures to be used to assure that user and group membership information will be kept current.	
Determine procedures to be used to create new user accounts and communicate initial passwords (e-mail, personal contact, other).	"Create a Password Strategy" on page 152
Decide if a simplified access approach is sufficient. This lets users log in based on operating system-level security.	"OS-Based Log-in Security" on page 138
Define how often users are required to changed passwords, and update the corresponding security setting.	"Expiration Days" on page 154
Define procedures for failed log-ins, 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 148 • "Monitoring System Security" on page 175
Define password policies and procedures, including password composition, length, expiration, and reuse of previous passwords.	"Create a Password Strategy" on page 152
Define appropriate policies and procedures for users requiring that sessions be locked using a screen saver or comparable mechanism whenever the user leaves the session unattended.	"Workstation-Level Security" on page 142

Table 12.2
Progress and Operating System Checklist

Topic	Reference
Determine whether to implement Progress as well as QAD user ID and password controls.	“Progress-Level Database Access” on page 141
Determine requirements for Progress-level schema security to control access to database tables.	“Progress-Level Database Schema Controls” on page 141
Consider disallowing Progress-level table and field access for the blank user ID	“Progress Editor Access” on page 140
Determine the period of inactivity after which a session should be disabled. For each device used to access the system, assure 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 session.	“Workstation-Level Security” on page 142
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 12.3
Security Parameters, Setup, and Processes Checklist

Topic	Reference
Verify and update relevant control program settings, especially those for security.	“Setting Up Security Control” on page 148
Review any currently defined users and groups and disable any inappropriate, inaccurate, or out-of-date entries.	“Controlling Access with User Groups” on page 163
Define users designated as security administrators, who will receive e-mail notification of security events such as failed log-ins exceeding a defined threshold.	• “Administrator Group” on page 151 • “Maximum Access Failures” on page 151
Update 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 logon attempts	“Create a Password Strategy” on page 152
Determine how security functions should be implemented to protect the integrity of database records. For each menu item, site, GL account, and so on, specify the appropriate users or groups authorized to execute the menu program or access data.	“Using Security Functions” on page 166
Review menu function authorizations for potential segregation of duty issues and adjust groups as appropriate.	“Controlling Access with User Groups” on page 163

Security Programs

Table 12.4 lists the menu programs you use in defining and maintaining security for your system.

Table 12.4
System Security Menu (36.3)

Number	Description	Program
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 Group Maintenance	mgurgpmt.p
36.3.5	User Group Inquiry	mgurgpiq.p
36.3.9	GL Account Security Maintenance	mgacsmt.p
36.3.10	Menu Security Maintenance	mgpwmt.p
36.3.11	Menu Security Change	mgpwcg.p
36.3.13	Entity Security Maintenance	glsecmt.p
36.3.14	Entity Security Inquiry	glseciq.p
36.3.15	Site Security Maintenance	clsismt.p
36.3.17	Inventory Movement Code Security	sosimt.p
36.3.18	Inv Mvmt Code Security Browse	gpbr502.p
36.3.19	Field Security Maintenance	mgflpwmt.p
36.3.20	Field Security by Group	mgflgpmt.p
36.3.22	User Access by Application Inquiry	lvusriq.p
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
36.3.23.4	User Group Report	mgurgprp.p
36.3.23.12	User Password Force Change Util	utfrcpsw.p
36.3.23.13	Entity Security Report	glsecrp.p
36.3.23.15	Site Security Report	clsisrp.p
36.3.23.16	GL Account Security Report	mgacsrp.p
36.3.23.19	Activated Field Security Report	mgflpwrp.p
36.3.23.20	Dictionary Field Security Report	mgfldcrp.p
36.3.24	Security Control	mgurpmmt.p

Setting Up Security Control

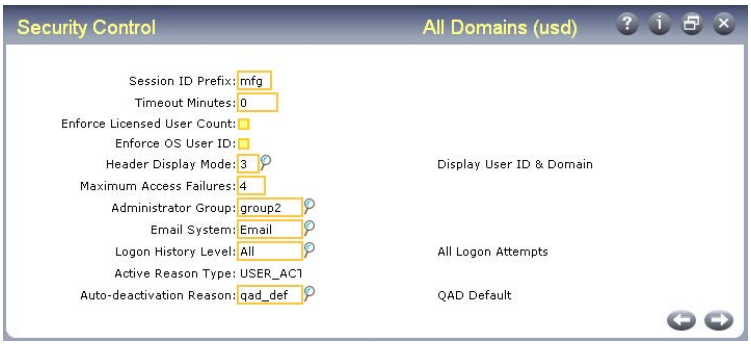
Use the two frames of Security Control (36.3.24) to:

- 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 members of the administrator group by e-mail. See page 154.
- 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 Progress Editor, to modify that record.

Fig. 12.6
Security Control (36.3.24), Initial Frame



Session ID Prefix. Enter a prefix for temporary system-generated work files. These are created in the directory where the user started the system. The default is TMP. Modify this field only if you access multiple databases from the same directory. If the prefix in both databases is TMP, one session's temporary files could overwrite another's.

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.

The field can also 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 142.

If you enter a value, the system considers a session inactive only when a menu is displaying. If the user is in a menu function—Item Master Maintenance, for example—a session is never automatically logged out.

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

No (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.

Yes: 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 log-in to the system
- When users attempt to use separately licensed applications or nonregistered applications

See "Registering Licenses" on page 85 for details on licensing.

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 and Windows sessions based on their operating system log-in.

No: Users are always required to enter a valid user ID and password.

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

See “OS-Based Log-in Security” on page 138.

Header Display Mode. Use this field to control the information that displays in the menu and program title bars of programs in the character and Windows user interfaces.

Note Display mode does not affect the display of programs in QAD Desktop or .NET UI.

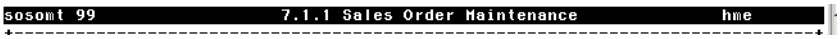
Valid values are:

0 (Display Date). The menu title bar displays the name associated with the ~SCREENS address code defined in Company Address Maintenance (2.12) and the current database name defined in Database Connection Maintenance. 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 12.7).

Fig. 12.7
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 log-in 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 log-in ID of the current user (see Figure 12.8).

Fig. 12.8
Display Mode 1



2 (Display Date with 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 12.9).

Fig. 12.9
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 log-in 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 log-in ID of the current user (see Figure 12.10).

Fig. 12.10
Display Mode 3

Note 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 and Windows interfaces, you can use the Ctrl+F key combination to review this information and other context details. In QAD Desktop, the user name displays by default in the browser title bar, along with the current domain and database name. In QAD .NET UI, it displays in the lower-right corner of the screen.

See “Using Ctrl+F to View Information” on page 17.

Maximum Access Failures. Enter the maximum consecutive failed log-in attempts allowed before the system deactivates the user’s log-in ID. When an account is deactivated, the system sends an e-mail message to members of the specified Administrator Group.

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

See “E-Mail Notifications” on page 154.

Note If you are using electronic signatures, this same value controls the number of failed signature attempts that are allowed before the system deactivates the user ID. See “Recording Electronic Signatures” on page 198.

Administrator Group. Designate a user group—defined in User Group Maintenance—as an administrator group. Group members receive e-mail notifications when specific security and controlled events occur; for example:

- When a user account is deactivated for too many failed log-in attempts. See page 154.
- If you are using audit trails, when an audit trail profile is activated or an error occurs during the audit trail creation process. See page 229.
- If you are using electronic signatures, when an electronic signature profile is activated or a user account is deactivated for too many failed signature attempts. See page 202.
- When an update is made in Security Control. See page 154.

Typically, this group includes the 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 members of the administrator group when security and Enhanced Controls events take place.

See “Building an E-Mail System Interface” on page 51.

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.

Important For system-generated e-mail to work correctly, be sure that the e-mail system definitions specified both here and for individual users are based on a message text *file*, rather than a message text *string*, in E-Mail Definition Maintenance.

Additionally, if you use the Windows user interface, the system uses the e-mail program on the client machine to send security-related e-mail. This means that a Windows e-mail program must be installed on each client machine. For example, if you use `wMailTo.exe`, that program must be installed and configured in the home directory on each client.

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

None (the default): Log-in history is not maintained.

Failed: Log-in history is maintained only for failed log-in attempts.

All: History is maintained for all log-in activity.

Particularly in highly regulated security environments, you can use log-in history information as part of an overall access monitoring effort. Use Logon Attempt Report (36.3.23.1) to view log-in history.

See “Monitoring System Security” on page 175.

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.

Active 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:

See “Using Reason Codes” on page 29.

- The Auto-Deactivation Reason field
- Reason codes entered manually in the Active Reason field in User Maintenance. See “Active Reason” on page 160.

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

- AUTO. The system automatically deactivated the account. You could enter this in Auto-Deactivation Reason.
- REACT. The system administrator has manually reactivated the account.
- NEW. The system administrator has added the account for a new user.
- LEFT. The user is no longer with the company, and the system administrator has deactivated 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 Active Reason field in the User Maintenance record of the default system user. 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-Deactivation Reason. Enter the reason code the system enters in user records when it automatically deactivates a user account. This occurs when the user reaches the number of consecutive failed log-in 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.

Create a Password Strategy

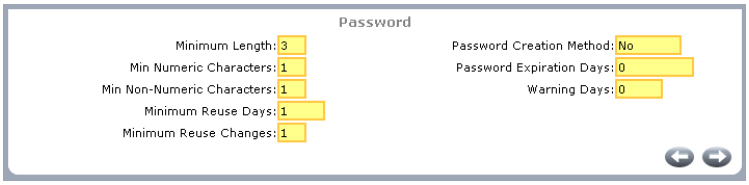
Use the Password frame to define the complexity requirements and expiration time period for user account passwords. Anytime a new password is created for an account—either manually or automatically—that password must meet the rules you set up here. Use as many or as few password parameters as required 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 system based directly on operating system security, do not define any password parameters; set Enforce OS User ID to Yes in the initial frame of Security Control. To default the user ID from the operating system but still require an application password at log-in, set that field to Yes and specify password parameters as needed.

See “OS-Based Log-in Security” on page 138.

Fig. 12.11
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 User Password Force Change Utility (36.3.23.12) to force users to change their passwords at the next log-in. New passwords must meet the updated structure requirements. See page 175.

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:

- 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.

See “Updating Passwords” on page 160.

- 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 “E-Mail Notifications” on page 154.

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

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 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 log-in and can optionally update their passwords immediately or, depending on menu access, update them in User Password Maintenance.

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 log-in attempts exceeds the number specified in Security Control, the system generates and sends e-mails to the specified administrator group. The e-mail text is similar to the following:

```
The purpose of this email is to inform you that a user has been deactivated 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 group identified in
Security Control.

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

This e-mail was automatically generated from a 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 Update Password is Yes 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 password. You have
been sent this e-mail because Security Control has been set up to e-mail autogenerated
temporary passwords.

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

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

- When any field is updated in Security Control, the system generates and sends e-mails to the specified administrator group. The e-mail text is similar to the following:

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

Changes made by user: jnw

Changed Field: old, new
======================
Administrator Group: 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 process. If you have any questions about
this email, contact the system administrator. Do not reply to this email.
```

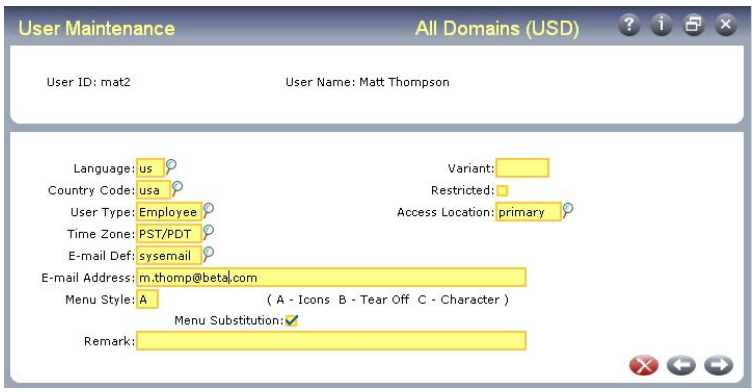
Note Values shown in this message are those stored in the database and may not be the same as displayed in the user interface. For example, the Administrator Group values display as the unique object identifier (OID) codes associated with the old and new values in the database. The message is intended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

Defining Users

You define users by assigning a unique ID in User Maintenance (36.3.1). Each program is always passed the user's ID, any group names associated with the user for the current domain, and access information associated with the user. After you create the ID for a user, you specify other identifying information and preferences.

Note User IDs cannot be blank, or the same value as a user group name.

Fig. 12.12
User Maintenance (36.3.1)



To log in, each user must specify a unique user ID and the associated password. Other user data is referenced throughout the system and may be required for reasons other than security.

See “Basic Login Security” on page 137.

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.

Note Batch processes must be assigned a valid user ID.

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

Note This restriction ensures a complete audit trail of users who have accessed the system.

Interaction with Licensing

There are three license types: named user, concurrent session, and business process cycle, which is used in combination with named user licensing. Concurrent session licensing is checked at log-in. Named user licensing is verified in User Maintenance.

For a named user license, the system counts the number of active users authorized to access the licensed application and compares the number against a predefined limit for the license.

If the number of active users exceeds the predefined limit specified in the license agreement, a violation message displays in User Maintenance. Violation messages can be either warnings or errors, depending on whether enforcement of the license policy is implemented. Errors prohibit actions in User Maintenance when the limit on license agreements is reached; warnings allow actions to continue.

See “Registering Licenses” on page 85.

System administrators can implement enforcement of license agreements by setting the Enforce Licensed User Count field to Yes in Security Control (36.3.24). Setting this field determines whether:

- Errors or warnings display in User Maintenance.

- System administrators can create new users when the number of existing users exceeds the licensed number.
- Additional users can log in when the number of sessions exceeds the licensed number.

The applications a user can access must be activated for the user. You can activate access to applications here or when you register an application license code in License Registration (36.16.10.1). This includes the base application. If a user ID is obsolete, you should inactivate access to all registered applications.

See “Setting Up Security Control” on page 148.

Controlling Information Process and Display

You can ensure that data is correctly displayed and processed for a given user—regardless of the user’s language or location—by specifying the following values in User Maintenance:

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.

Country Code. Enter a three-character country code to associate with the user. The country code must be defined in Country Code Maintenance (2.14.1) and it must have an associated alternate country code.

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

See *User Guide: Master Data.*

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 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

Use the following fields to identify this user:

User Type. Enter 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.

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 may 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 (36.4.3) 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). See “Setting a Default Time Zone” on page 14.

Remark. Use this field to 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 deactivated.

Specifying E-Mail Addresses

Associate a valid e-mail address and definition with each user who receives messages generated by the system.

See “Building an E-Mail System Interface” on page 51

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

- System administrators can receive automatic notification when user IDs are deactivated because of log-in 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.
- Enhanced Controls uses e-mail to inform administrators of unusual audit trail and electronic signature events.

Setting Interface Preferences

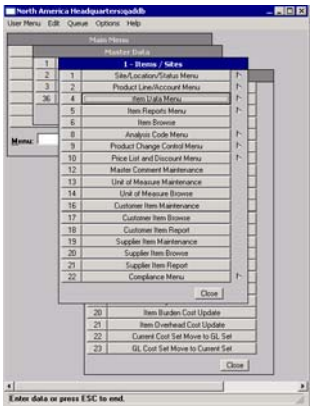
Select interface preferences for individual users by specifying values for the following:

- Whether menu substitution is enabled or disabled. This only affects menus in Windows and character sessions. This does not affect the QAD Desktop and .NET interfaces.
- Menu style. This only affects the menu style used in a Windows session.
  - The icon menu style has large buttons that lead you into the different parts of the system and show a hierarchy of your location in a submenu.
  - The tear-off menu style enables you to choose your menu layout.
  - The character-based menu style emulates traditional character terminals.

Fig. 12.13
Menu Style Options (GUI Interface Only)

Character Menu

Tear-Off Menu



Icon Menu

Specifying Security Settings

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

Fig. 12.14
User Maintenance, System Access Frame

Active. Indicate whether this user ID can be used to log in to the system. To deactivate an existing user ID, enter No.

Note Anytime this field is updated, the Active Reason field must also be updated.

Active is updated in the following ways:

- Automatically when you enter a new user ID. By default, the system sets Active to Yes; you must manually enter an active reason.
- Automatically when the system deactivates an account for too many failed log-in attempts. Active Reason is set to the code specified in Security Control. See “Maximum Access Failures” on page 151.
- Manually when you update an existing ID; for example, you can do this to reactivate a system-deactivated user record, or to deactivate an account when a user leaves the company. You must enter an active reason.

Inactive accounts do not display in most user lookups.

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

Active Reason. Enter a reason code that indicates the reason for modifying the setting of Active. This reason code must be associated with reason type USER_ACT.

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

See “Active Reason Type” on page 152.

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.

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.

The default is Yes for new users and cannot be updated. This lets you assign temporary, single-use passwords either automatically or manually.

The field defaults to No for existing users unless the password has been changed. In that case, it is set to Yes and you cannot update it. This forces users to assign their own passwords at the next log-in.

Use Force Password Change Utility (36.3.23.12) to set this field to Yes for selected users or user groups.

Update Password. Specify whether this user requires a new password. For new users, the field defaults to Yes and you cannot change it.

Updating Passwords

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

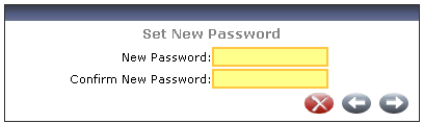
- Display. The system-generated password displays at the bottom of the screen.
- Email. The system generates a password and e-mails it to the user.

- No. Automatic password generation is disabled. A frame displays for you to manually enter a new password.

See “Create a Password Strategy” on page 152.

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

Fig. 12.15
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.

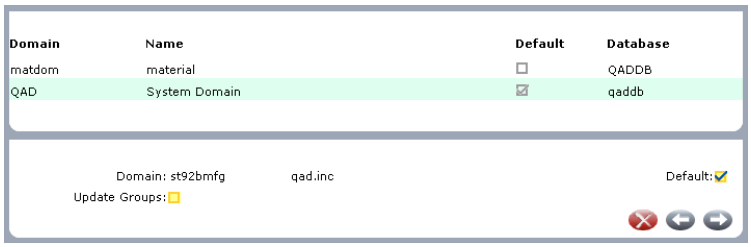
Passwords expire based on the value of Expiration Days in Security Control. If you want to let users change their own passwords at a time other than log-in, give them access to User Password Maintenance (36.3.3).

See “Expiration Days” on page 154.

Specifying Domains

Use the domain frame to enter or update the domains to which this user has access. If you specify more than one domain, identify the one the system uses as the default at log-in. Additionally, you can enter or update the user groups that control security access for the user while in this domain.

Fig. 12.16
User Maintenance, Domain Frame



Domain. Enter the code identifying a domain this user can access. The domain name displays next to the code.

The functions that the user can execute in this domain are determined by access granted to the groups associated with the user in this domain. Set Update Groups to Yes to enter or update the list of user groups associated with this user in this domain.

Default Domain. Enter Yes if this is the user’s default domain; otherwise, enter No. This field defaults to Yes for the first domain assigned to a user.

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 a QAD database, the system retrieves the information associated with the user in User Maintenance. In the character and Windows UIs, 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 log-in but is automatically logged in to the single domain associated with the ID specified.

Only one domain can be designated as default. When you enter Yes, the system verifies if another default domain exists for the user. If it does, a warning displays and you are prompted to continue. If you choose to continue, the current domain becomes the default and the system no longer uses the other domain as the default during log-in.

You cannot exit this frame without assigning a default domain to the user. An error is generated and you are prompted to continue. If you continue, any changes made in the current session are discarded.

A user with access to multiple domains can use Change Current Domain (36.10.13) to switch to another domain at any time during a session. Otherwise, all session activity takes place in the domain specified at log-in.

See “Changing the Current Domain” on page 13.

Update Groups. Enter Yes to display a frame that lets you enter or update the list of user groups associated with this user in this domain. Groups are required only if you control security access by group. Set this field to No to skip the group frame for this domain.

Note You also can associate users with groups in each domain in User Group Maintenance (36.3.4). See “Defining User Groups” on page 164.

Specifying User Groups

Group membership can determine whether a user is given access to menus, sites, and other system elements. The system always considers the user’s ID and any group names associated with the user for the current domain when allowing access to various functions.

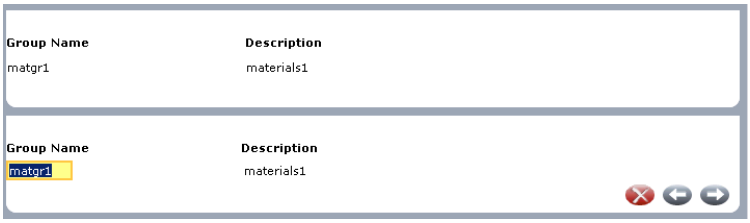
Use groups to streamline security setup. Many users can belong to a single group; when a new user record is created, you can add the user to existing groups to ensure they have correct access.

When Update Groups is Yes in the domain frame, the system displays a list of all the groups currently assigned to the user for the domain. Use the Group Name field to add groups, or press Delete to remove them.

See “Controlling Access with User Groups” on page 163.

Note Groups must be defined in User Group Maintenance (36.3.4) before you can enter them here.

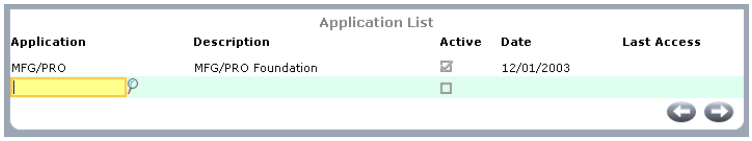
Fig. 12.17
User Maintenance, Group Frame



Specifying Application Use

The Application List frame in User Maintenance lets you define the software applications that a user can access. When you define a new user, the system prompts you authorize the new user for all licensed applications. If you respond Yes, Active is set to Yes for all licensed applications for this user. Otherwise, QAD Enterprise Applications is listed as the only active application. You can list additional licensed software applications, then set Active to Yes or No for each application. The default is Yes.

Fig. 12.18
User Maintenance, Application List Frame



The application name you enter under Application Name must be registered through License Registration (36.16.10.1).

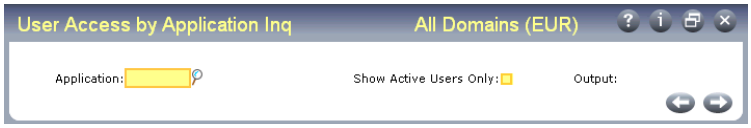
You can also specify which users can access an application after you register the application in License Registration.

See “Interaction with Licensing” on page 156.

If you deactivate the system for a user, all other registered applications are deactivated, too.

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. 12.19
User Access by Application Inquiry
(36.3.22)



Controlling Access with User Groups

You can assign users to groups, then control access to various system elements based on group membership. This feature provides flexibility and consistency in the way you enforce security requirements.

Important A group security feature is also available in QAD Desktop. However, this is supported by a different set of records than standard user groups. See *User Guide: QAD Desktop* for information.

First, define groups in User Group Maintenance (36.3.4). Then use the following programs to assign access based on groups:

- Menu Security Maintenance (36.3.10). See page 168.
- Site Security Maintenance (36.3.15). See page 172.
- GL Account Security Maintenance (36.3.9). See page 174.

- Inventory Movement Code Security (36.3.17). See page 175.

When a user is given access to more than one domain, you can use groups to manage roles within the domain. See “Specifying Domains” on page 161.

Example A user has access to all functions in Domain1, but can only generate reports in Domain2. Assign the user to two groups: Admin and Review. Set up menu security so that Admin can access all functions and Review can access only reports. Then, in User Maintenance, assign the user to the Admin group for Domain1 and to the Review group for Domain2.

Grouping users reduces maintenance for the system administrator.

For most security, the use of groups is entirely appropriate. In a few cases, you might not want to use groups. For high-risk functions such as Menu Security Maintenance, grant access to specific users by ID—typically the system administrator and an alternate.

Defining User Groups

Use User Group Maintenance (36.3.4) to create groups that can be used to control access to various aspects of system use and associate them with domains and users.

Two administrative user groups are required in the system:

- The administrator group specified in Security Control (36.3.24) to receive e-mail notifications when specific security and controlled events occur. See “Administrator Group” on page 151.
- A QAD Desktop administrative group that can access the Desktop administrative functions. This group is specified during installation of QAD Desktop 2.7 or higher.

  Note If you plan to take advantage of the simplified screen tool in QAD Desktop 2.7, you can also specify a user group that can create and modify screen templates, which are then assigned to groups of users. See *User Guide: QAD Desktop* for details on simplified screens.

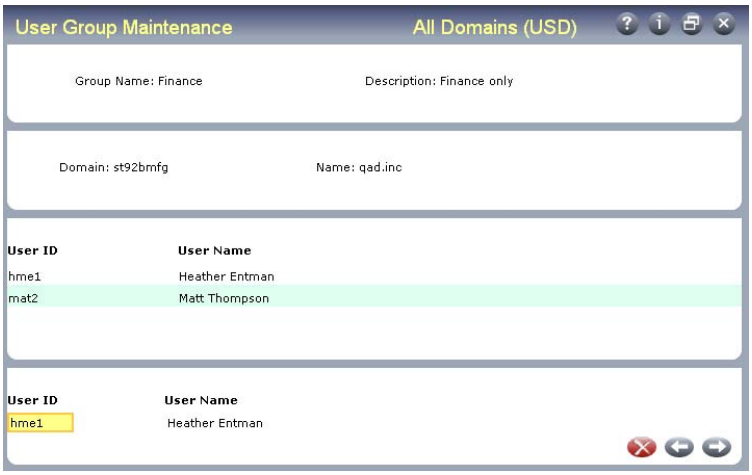
Although they can streamline security setup and administration activities, groups are not required to control access. Depending on your security requirements, you can also control access based on individual IDs or not at all.

You also can assign user IDs to existing groups by domain in User Maintenance (36.3.1). To use this method, just set up a group name and description in User Group Maintenance, and set Update Groups to Yes in the domain frame of User Maintenance.

Use the first frame to enter a name and description for the group. Then specify the domain with which you want to associate group records. The domain must be defined in Domain Maintenance (36.10.1), and the Active field must be Yes in that program.

See “Specifying User Groups” on page 162.

Fig. 12.20
User Group Maintenance (36.3.4)



The system lists all users currently assigned to this domain/group combination. To add a user ID, navigate to the User ID field in the bottom frame and enter a user ID defined in User Maintenance. If the user is not currently assigned to the domain, User Group Maintenance automatically creates that association.

If you enter a deactivated user ID—one that has Active set to No in User Maintenance—the system displays a warning message. Although the user ID is considered part of the group, the user cannot log in to the system until the user ID is reactivated.

See “Active” on page 159.

Deleting Group Records

To delete a user ID from the group, select the ID from the list and choose Delete. Confirm the delete to continue.

To delete a group, you must first delete all the domain-specific group records. Navigate to the Domain field and choose Delete. When you confirm the deletion, the system removes all references to the group from access lists associated with each domain. After deleting these records, the system prompts you to delete the group itself.

Note Deleting a group has no effect on access records set up using Field Security by Group (36.3.20). That program creates an individual access record for each group member; these records are not updated when a group is deleted. You must use Field Security Maintenance (36.3.19) to delete records individually by user ID. See “Field Security by Group” on page 171.

User Group Example

In this example, the system administrator employs user groups and menu security to control access to three functions based on each employee’s organizational level.

Company A wants to provide three levels of access to accounts payable (AP) functions: one for clerks, one for managers, and one for the CFO.

The system administrator creates three groups in User Group Maintenance: *Clerk*, *Manager*, and *CFO*. Sara, the AP Clerk, is added to the Clerk group. Don, the AP Manager, is added to the Manager and Clerk groups. Helen, the CFO, is added to all three groups. In this setup, Helen's group membership grants her entry to all the levels she is authorized to access.

Fig. 12.21
Using Groups to Give Access

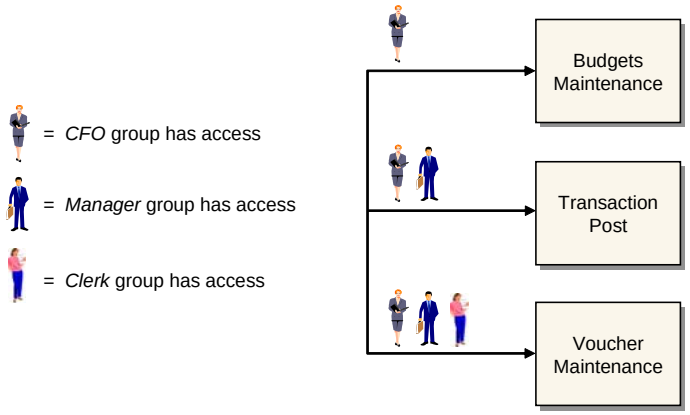


Table 12.5 shows how the system administrator sets up user access for each group in User Group Maintenance.

Table 12.5
Sample Group Setup

Group	User
Clerk	Sara
Manager	Don
CFO	Helen

Next, the administrator uses Menu Security Maintenance (36.3.10) to grant each group access to the appropriate programs.

When Mark is hired as the new deputy CFO, the system administrator only has to assign Mark to the *CFO* group—rather than using Menu Security Maintenance to give the new user access to each individual protected AP function.

Using Security Functions

You can use individual user records as well as user groups to limit system access based on the following:

- The menu system
- Individual fields
- Sites
- General ledger accounts
- Inventory movement codes

Note Except for menu security, access records apply only to the current domain from which they are entered.

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

Specifying Groups or Users

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

- Menu Security Maintenance (36.3.10). See page 168.
- Site Security Maintenance (36.3.15). See page 172.
- Inventory Movement Code Security (36.3.17). See page 175.
- GL Account Security Maintenance (36.3.9). See page 174.
- Entity Security Maintenance (36.3.13), which is based only on user ID access. You cannot assign user groups. See page 173.

Note If you do not set up records in these programs, the system by default allows access to all users who pass log-in and domain security restrictions. See "Basic Login Security" on page 137.

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

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

- The asterisk (*) gives access to all users and groups.
- The exclamation point restricts specific users by user ID, not by group. For example, `!user1,*` means all users except user1 have access to the function; `!user1,admin` allows access only to members of the admin group, with the exception of user1. However, `!admin,*` does not prevent members of the admin group 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, groups, 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 groups. The asterisk allows access to all group members, even if they have been excluded as individuals.

Table 12.6 lists some examples. User IDs and group names are not case-sensitive.

Table 12.6
Sample Uses of User ID and Group Name

String	Description
*	All users have access.

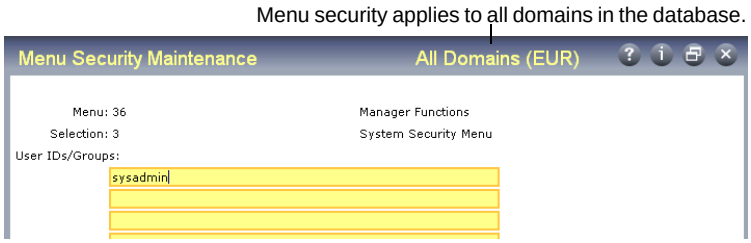
String	Description
`mary, manager`	Only user mary and members of the manager group have access.
`!jcd,*`	Everyone but user jcd has access.

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.

Assign Access by Menu

Menu security controls each user's access to programs. Use Menu Security Maintenance (36.3.10) to define the users or groups that have access to a menu function.

Fig. 12.22
Menu Security Maintenance (36.3.10)



Specifying Menu Numbers

Enter the number identifying the menu where the program you want to restrict is found. The system combines menu and selection number to determine the function to be restricted.

For example, Purchase Order Maintenance is selection 7 on menu 5.

To restrict an entire submenu, enter the menu number in Menu and the submenu in Selection. For example, Menu 7 Selection 1 restricts the entire Sales Order Menu (7.1).

The main menu is identified by menu number 0 (zero). In the character interface, restrict access to any of the 36 top-level menu items by specifying 0 for menu and the appropriate number for selection.

In the Windows environment, top-level menu options are also grouped under seven icons. These icons are referenced through the letter A:

A.1: Distribution

A.2: Manufacturing

A.3: Financials

A.4: Customer Services

A.5: Master Data

A.6: Custom

A.7: Supply Chain

To restrict access to top-level menu items in the Windows interface, define records for menu A and the appropriate selection number as well as menu 0 and the appropriate selection.

For example, to restrict access to the Item/Sites menu (1) in Windows, create the following records:

1 Specify Menu: 0, Selection 1.

2 Specify Menu A.5, Selection 1.

Important Menu security is intended to provide control over menu functions as opposed to the executable programs associated with a particular menu specification. This distinction is important in cases where a particular function is provided in multiple menu locations. For example, AR Aging as of Effective Date (`arcsrp05.p`) exists at the following menu locations:

- 26.21.1.12.16
- 26.21.3.1.16
- 27.18

If User1 is denied permission to execute this function from the first two menu locations, this user may still be able to execute the function from the third location.

Effect of Menu Security

The effect of menu security varies according to the interface.

- In the character interface, users cannot see restricted menu items or submenus.
- In the Windows interface, a restricted menu item or submenu displays with an X after the menu number. Users can choose not to see restricted menus by selecting Hide Menu Items from the Options menu.
- In QAD Desktop, restricted menus display but users cannot execute them.
- In QAD .NET, restricted menu items are hidden.

In all interfaces, users cannot access a restricted menu item by typing the program name. However, programs can still be executed from the Progress editor unless you add security for it. To do this, leave Menu blank in Menu Security Maintenance and specify selection 1, which represents the editor.

See “Progress Editor Access” on page 140.

Limit Access to Fields

Field security prevents unauthorized users from updating secured fields. It does not prevent them from seeing the value of a field if they have access to the screen where it is updated. Neither does it protect a field from program-level updates through custom code.

The system determines whether a user is authorized based on whether the user ID matches the values specified for the field. User groups are supported through a two-step process.

See “Specifying Groups or Users” on page 167.

Field Security Validation

In the standard release, security is not active for any fields, and only a few fields are eligible for field security. Use the Dictionary Field Security Report (36.3.23.20) to determine which fields can be given security.

In the character and Windows interfaces, you can also 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. The expression must reference `gppswd.v`. The syntax is:

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

Activated Field Security Report

Use the Activated Field Security Report (36.3.23.19) to see which fields have security activated. It also lists privileged user IDs.

Dictionary Field Security Report

The Dictionary Field Security Report (36.3.23.20) 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 groups. 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 (36.3.19).
2. Verify that the field is secured by running the Activated Field Security Report (36.3.23.19).

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 (36.3.19).

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 you want to add security to.

   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 Go, 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 (36.3.19).

9 Verify that each field is secured by running the Activated Field Security Report (36.3.23.19).

Note For multi-language implementations, you must run `mgfldcmt.p` in the base language instance. Then you must recompile your code for the base language and all other languages you have implemented.

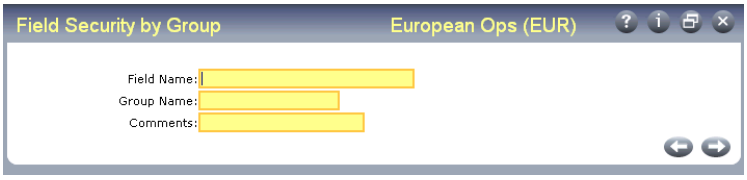
Field Security by Group

You can also set up field security for a group of users.

1 Assign users to groups in User Group Maintenance (36.3.4) or User Maintenance (36.3.1).

2 Execute Field Security by Group (36.3.20). This function adds all users who belong to a specified group to the list of authorized users for a validated field.

Fig. 12.23
Field Security by Group (36.3.20)



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

- If you remove a user from a group that was given access to a field, that user can still access the field. To prevent this, use Field Security Maintenance (36.3.19) to remove the individual user.
- You cannot use Field Security by Group to remove a group of users from the list of authorized users. To remove a group, you must remove every individual in the group in Field Security Maintenance.

- If you delete a group in User Group Maintenance, individual records remain on the system until you delete them in Field Security Maintenance.

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

Control Inventory 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.

Access is managed by user and by group. A user can access a site only if that user's ID or group name appears in the Groups field in Site Security Maintenance (36.3.15).

Fig. 12.24
Site Security Maintenance (36.3.15)

When a user enters a restricted site code in a site-controlled program, the system checks the value of the Groups field associated with the site in Site Security Maintenance. If the user does not belong to an associated group, or 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

Because of the complexities of security, it is important to plan site security carefully and to follow closely the procedures for creating user and group names and associations. Users who are not listed individually or who have no group memberships in Site Security Maintenance (36.3.15) cannot complete transactions at secured sites.

To implement site security, associate groups with users in User Maintenance or User Group Maintenance.

See “Specifying Groups or Users” on page 167.

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.

Control Entity Access

When entity security is in place, only authorized users can complete update transactions in the General Ledger module for particular entities. Update transactions include:

- Transaction maintenance
- Posting
- Consolidation
- Export and import
- Budget maintenance
- Opening and closing fiscal periods

Entity security also affects the following programs in other modules:

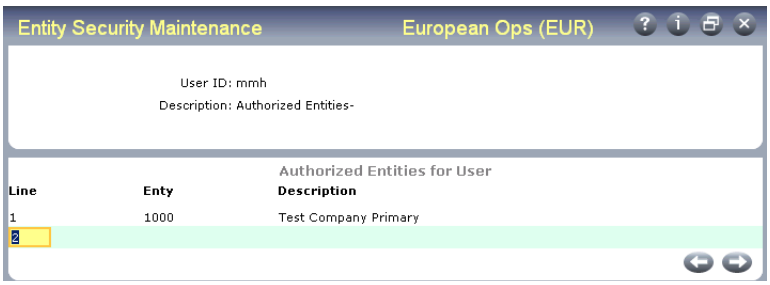
- Debit/Credit Memo Maintenance (27.1)
- Payment Maintenance (27.6.4)
- Voucher Maintenance (28.1)
- Voucher Confirmation–Automatic (28.6)
- Voucher Confirmation–Manual (28.7)
- Cash Book Maintenance (31.13)
- Fixed Asset Transaction Post (32.13)
- Fixed Asset Transaction Void (32.14)
- Fixed Asset Retirement (32.19)

All users can still enter maintenance functions or run inquiries and reports. To control access to a screen, you must use menu security.

Once you define entity security for one user and one entity, it applies to all users and entities. Each user must be set up individually. To give a user access, enter the user ID and list of entities, as shown in Figure 12.25.

Note Entity security cannot be defined for groups.

Fig. 12.25
Entity Security Maintenance (36.3.13)



An asterisk in the Entity field indicates that a user can access all entities.

Important For a user to create a new entity, they must have access to all entities (*).

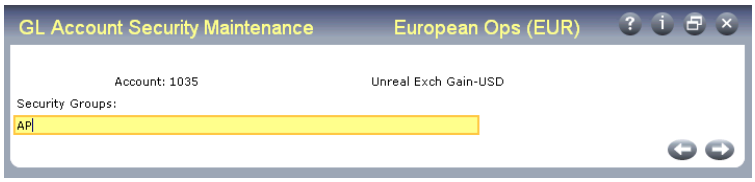
Define GL Account Security

GL account security lets you restrict who can update GL accounts based on user ID or user group. Account security is only effective when Verify GL Accounts is Yes in Domain/Account Control (36.1).

Use GL Account Security Maintenance (36.3.9) to assign users or security groups to account numbers. Use the GL Account Security Report (36.3.23.16) to list all accounts that have controlled access.

See “Specifying Groups or Users” on page 167.

Fig. 12.26
GL Account Security (36.3.9)



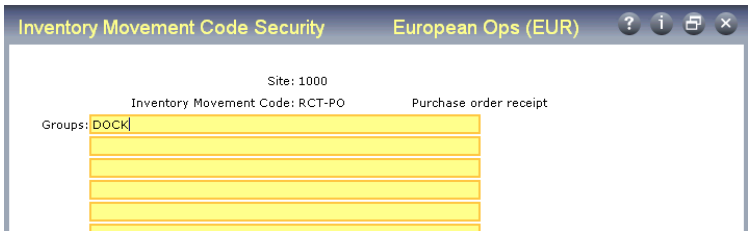
When a user attempts to create a transaction affecting an account, the system checks the user ID and the groups associated with the user against the list associated with the account. If a match is not found, a message displays and the user cannot complete the transaction.

Note Account security is not applied during Transaction Post. Use Menu Security Maintenance (36.3.10) to restrict posting functions.

Define Inventory Movement Code Security

Use Inventory Movement Code Security (36.3.17) to grant or deny access to individuals and groups to shipping transactions that reference a specific inventory movement code at a particular site.

Fig. 12.27
Inventory Movement Code Security (36.3.17)



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: Distribution.*

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 (36.3.18) to display inventory movement code security records. Fields associated with a record can be viewed 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.

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 reactivated it.
- When an account is deactivated, the e-mail system can automatically notify members of the administrator group. 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. User Password Force Change Utility (36.3.23.12) lets the administrator force users to update their passwords based on user group, domain, and/or the date of the last change.

- Depending on the level of log-in history specified in Security Control, use Logon Attempt Report (36.3.23.1) to track when log-in 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 log-in, as well as the date, time, server time zone, and other data relevant to the log-in event.

  Note If you are using electronic signatures, E-Signature Failure Report (36.12.7) lets you monitor unsuccessful signature events. See page 204.

  Example You can set up batch processing to run this program each morning to identify all failed log-in attempts on the previous day.

- Each time a user account is activated or deactivated, the Active Reason Code field in User Maintenance must be updated. This happens automatically when an account is deactivated as a result of excess unsuccessful log-in attempts. Otherwise, the administrator must enter a reason code manually.

Chapter 13

Electronic Signatures

This chapter discusses the following topics:

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 module 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 program that records signatures can sign records. Use Menu Security (36.3.10) to assign access to signature-controlled functions based on user groups or individual user IDs. See “Assign Access by Menu” on page 168.

These features are intended as part of an overall approach—also incorporating capabilities offered by System Security and Audit Trails—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. Table 13.1 lists the programs that currently can have electronic signatures enabled.

See “Profiles” on page 183.

Table 13.1
Programs Included in Default Profiles

Module	Menu	Program
Product Change Control (PCC)	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
Compliance	1.22.1	Lot Master Maintenance
	1.22.24	Compliance Control
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
	3.24	Inventory Control

Module	Menu	Program
Shop Floor Control	17.1	Labor Feedback by Work Order
	17.2	Labor Feedback by Employee
	17.3	Labor Feedback by Work Center
	17.4	Non-Productive Labor Feedback
	17.5	Operation Complete Transaction
	17.6	Operation Move Transaction
Quality Management	19.11	Quality Order Results Entry
	19.13	Test Results Maintenance

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 207.

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 Work Flow

Use the programs on the E-Signature Setup Menu (36.12.14) to set up and configure electronic signature functions. Figure 13.1 illustrates the electronic signature process work flow; use it to set up signature functions in your environment.

Fig. 13.1
Electronic Signatures Setup Flow

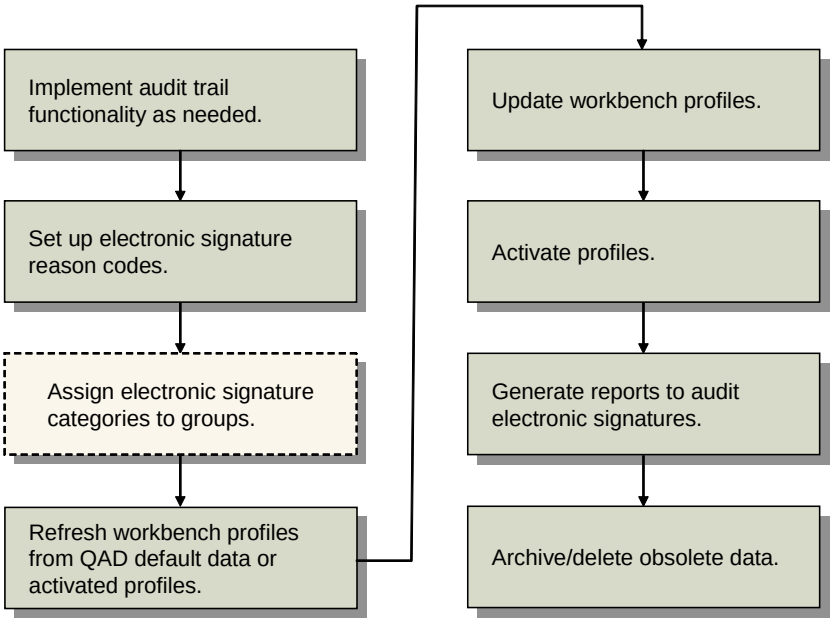


Table 13.2 shows the programs available for setting up and reporting on electronic signature functions.

Table 13.2
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
36.12.14.21	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 the system fits 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 audit trail and security requirements for signed records; for example, assign menu 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 QAD Enterprise Applications 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. In the system, the reason code provides the signature meaning. Be sure to plan and implement reason codes that make sense in your specific regulatory environment.

See page 187.

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 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 189.

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 197.

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 204.

Categories

A category is a QAD-provided definition of a set of 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 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 17) 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 186.

Table 13.3 lists the electronic signature categories, as well as the default menu programs associated with them. If for some reason 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 195.

Table 13.3
QAD-Defined Categories

Code	Name	Description	Available Menu Programs
0001	InvCtrl	Inventory Control	Inventory Control (3.24)
0002	QualRes	Quality Results	Labor Feedback by Work Order (17.1) Labor Feedback by Employee (17.2) Labor Feedback by Work Center (17.3) Operation Move Transaction (17.6)
0003	OpHist	Operation History	Labor Feedback by Work Order (17.1) Labor Feedback by Employee (17.2) Labor Feedback by Work Center (17.3) Non-Productive Labor Feedback (17.4) Operation Complete Transaction (17.5) Operation Move Transaction (17.6)
0004	ComCtrl	Compliance Control	Compliance Control (1.22.24)
0005	LotMstr	Lot Master	Lot Master Maintenance (1.22.1)
0006	InvDet	Inventory Detail	Inventory Detail Maintenance (3.1.1) Detail Maintenance by Item/Lot (3.1.2)

Code	Name	Description	Available Menu Programs
0007	InvTran	Transaction History	Inventory Detail Maintenance (3.1.1) Detail Maintenance by Item/Lot (3.1.2) 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) Test Results Maintenance (19.13)
0009	PCOInc	PCO Incorporation	Incorporation Selection (1.9.7.4) Incorporation (1.9.7.5) Implementation (1.9.7.13)
0010	PCOAppr	PCO Approval	PCR/PCO Approval (1.9.6.1) Detail Approval Maintenance (1.9.6.13)

Note Some categories are also associated with reports and inquiries that can include electronic signature data. See “Functional Reports and Inquiries” on page 207 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 13.3 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 198.

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

Profiles

The electronic signature system maintains signature information based on a signature profile that is associated with a specific category of data. 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

Note Profiles are identified by the corresponding QAD-defined category codes.

The life cycle of a profile consists of three phases:

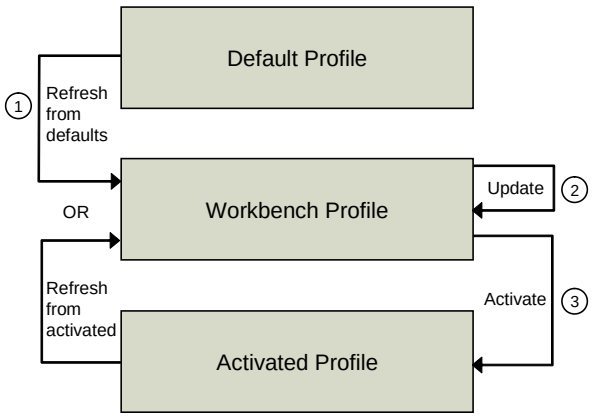
- The QAD-provided default profile. Based on QAD-provided category data, this is loaded when you install Enhanced Controls and serves as the template for profiles used by the system. You cannot update default profile records directly—only after you have copied them by refreshing the workbench profiles.

  See “Refreshing Signature Profiles” on page 190.

  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 192.
- 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 197.

Figure 13.2 summarizes the relationships between the three category profile types.

Fig. 13.2
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.

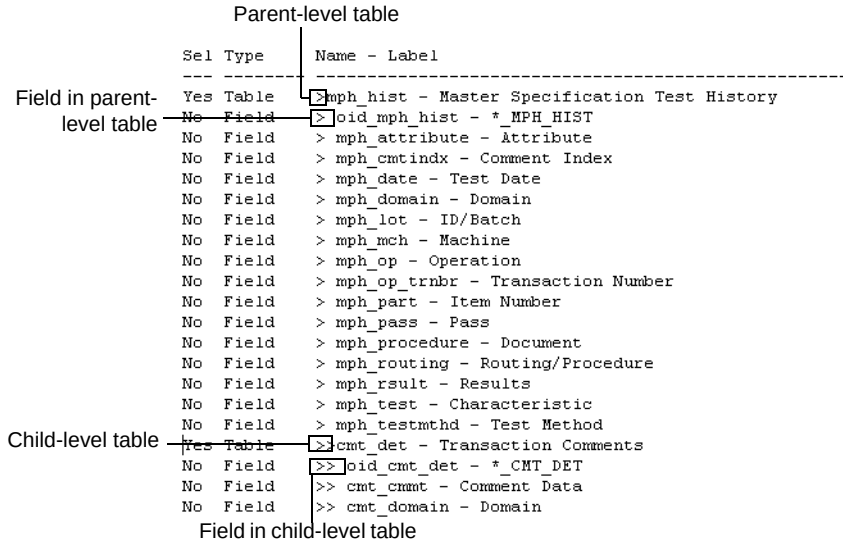
Each category profile includes one or more database tables and their corresponding set of fields. For example, the profile for category 0007, Transaction History, includes fields from the inventory transaction history table (tr_hist). In some cases, a category profile might include multiple tables where 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.

See “Updating Signature Profiles” on page 192.

Example Figure 13.3 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 17). View default profiles using E-Sig Workbench Profile Report (36.12.14.6) with Display Default Profiles set to Yes.

Fig. 13.3
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 204.

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

Table 13.4
Available Filters, by Category

	Filter				
Category	Domain	Site	Item Number	Location	Work Center
0001 Inventory Control	✓				
0002 Quality Results	✓	✓	✓		✓
0003 Operation History	✓	✓	✓		✓
0004 Compliance 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.

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.

See “Set Up Filters” on page 196.

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

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.

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:

- Set up audit trail functionality
- Define signature reason codes
- Check Security Control settings

Set Up Audit Trails

Although electronic signature functions can be used without Audit Trails—signature data is stored in production database tables rather than in the audit database—this is not a typical business case. Signatures normally fill only part of the user accountability requirements of a regulated environment. Other important elements are:

- Access security control and tracking, provided by the System Security module
- The ability to identify changes to the database, as well as identify who made them—the primary function of Audit Trails

Additionally, as part of the overall accountability process, electronic signature records cannot be deleted unless they are first archived to an audit database. If you ever want to delete signature records, you must have at least one audit database in place and connected.

For the audit database you will use for signature archive, set E-Signatures to Yes and specify an associated begin date in Audit DB Maintenance (36.12.13.11). This is the database used for archiving electronic signature records. Whether you use the same database that stores audit trail records should be determined as part of your overall audit database planning.

See “Setting Up Database Connections” on page 217.

Define 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. In the system, the reason code provides 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.

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

See “Recording Electronic Signatures” on page 198.

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.

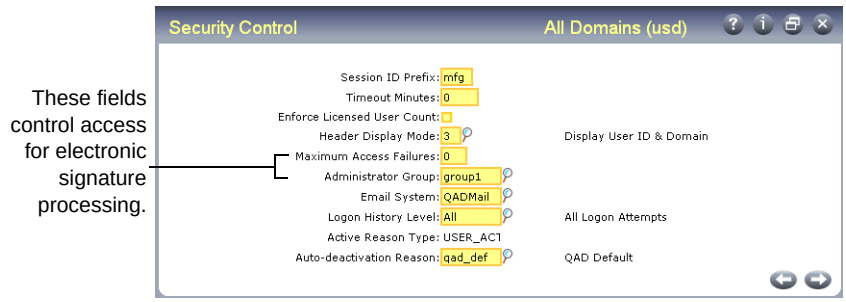
Review Security Control Settings

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 log-in process.

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

See Chapter 12, "Users and Security," on page 133 for information on setting up and using log-in security.

Fig. 13.4
Security Control (36.3.24)



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

- Max Access Failures indicates how many consecutive unsuccessful signature attempts cause the user's session to terminate, deactivate the account, and inform the administrator group of a potential unauthorized access attempt.
- Administrator Group is the name of the user group—defined in User Group Maintenance (36.3.4)—whose members are notified by e-mail when a session is terminated because of excessive unsuccessful signature attempts. The system also sends e-mail to this group when a signature profile is activated. See "E-Mail Notifications" on page 202.

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 define 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. 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.

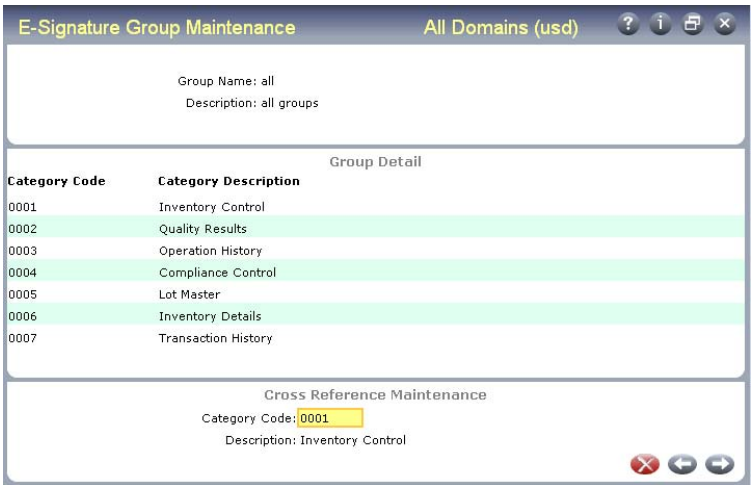
See “Profiles” on page 183.

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.

Fig. 13.5
E-Signature Group Maintenance (36.12.14.1)



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 Go 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.

Fig. 13.6
E-Signature Workbench Refresh (36.12.14.4)

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.

Note If Value is blank, the system prompts you to confirm.

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

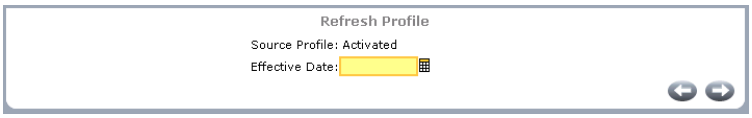
Refresh Profiles. Indicate whether to refresh all data for the specified profiles. When this field is Yes, an additional frame displays that you can use to determine which profiles are used as the source of the updates.

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.

Fig. 13.7
E-Signature Workbench Refresh, Refresh Profile Frame



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.

Fig. 13.8
E-Signature Workbench Refresh, Override Fields Frame

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 197.

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 183 for information.

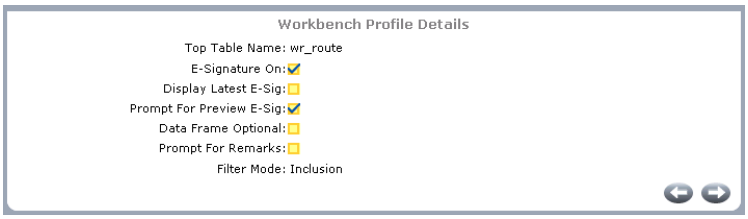
Specifying Control Settings

Figure 13.9 illustrates the first frame of E-Signature Workbench Profile Maintenance.

Fig. 13.9
E-Signature Workbench Profile Maintenance (36.12.14.5)

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

Fig. 13.10
E-Signature Workbench Profile Maint, Workbench Profile Details



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 184.

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.

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

Figure 13.16 on page 199 shows the frame that displays when this field is Yes.

See “Recording Electronic Signatures” on page 198.

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.

Note 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.

Figure 13.17 on page 200 shows the frame that displays when this field is Yes.

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 when the user interacts 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 take a break 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.

See Figure 13.17 on page 200.

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.

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.

See “Filters” on page 186.

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 13.5 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 13.5
Profile Precedence for Multiple Categories

Menu Program	Category Sequence
Labor Feedback by Work Order (17.1) Labor Feedback by Employee (17.2) Labor Feedback by Work Center (17.3) Operation Move Transaction (17.6)	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.

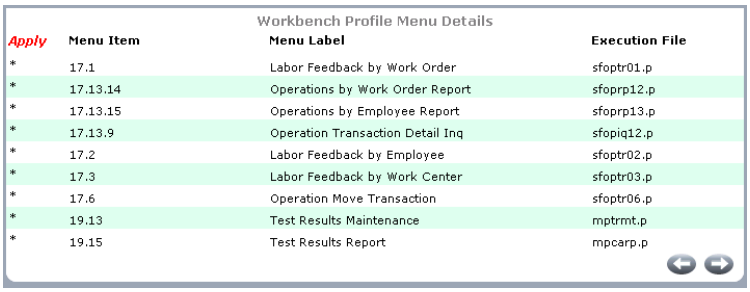
When a program is included in the category profile, an asterisk (*) displays in the Apply column. Clear the field to deselect a program.

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. 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.

See “Functional Reports and Inquiries” on page 207.

Fig. 13.11
E-Signature Workbench Profile Maint, Workbench Profile Menu Details



Workbench Profile Menu Details

Apply	Menu Item	Menu Label	Execution File
*	17.1	Labor Feedback by Work Order	sfoptr01.p
*	17.13.14	Operations by Work Order Report	sfoprp12.p
*	17.13.15	Operations by Employee Report	sfoprp13.p
*	17.13.9	Operation Transaction Detail Inq	sfopiq12.p
*	17.2	Labor Feedback by Employee	sfoptr02.p
*	17.3	Labor Feedback by Work Center	sfoptr03.p
*	17.6	Operation Move Transaction	sfoptr06.p
*	19.13	Test Results Maintenance	mptrmt.p
*	19.15	Test Results Report	mpcarp.p

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.

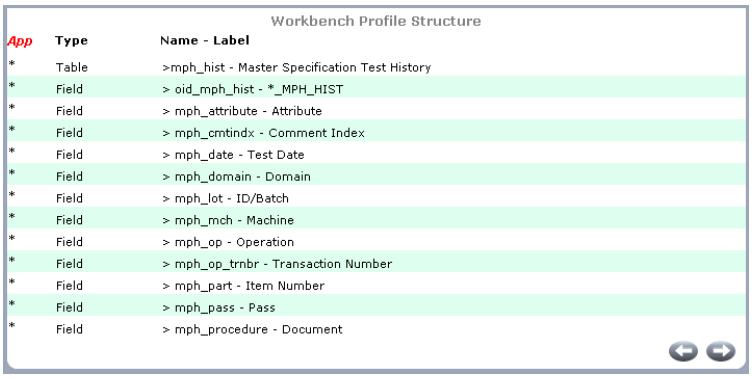
Toggle the asterisk in the Sel column to select or deselect fields or tables. If you deselect or select a table, all fields in the table are automatically deselected or selected as well. In that case, the frame display does not refresh immediately.

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 184.

Fig. 13.12
E-Signature Workbench Profile Maint, Workbench Profile Structure



Workbench Profile Structure

App	Type	Name - Label
*	Table	>mph_hist - Master Specification Test History
*	Field	> oid_mph_hist - *_MPH_HIST
*	Field	> mph_attribute - Attribute
*	Field	> mph_cmtindx - Comment Index
*	Field	> mph_date - Test Date
*	Field	> mph_domain - Domain
*	Field	> mph_lot - ID/Batch
*	Field	> mph_mch - Machine
*	Field	> mph_op - Operation
*	Field	> mph_op_trnbr - Transaction Number
*	Field	> mph_part - Item Number
*	Field	> mph_pass - Pass
*	Field	> mph_procedure - Document

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.

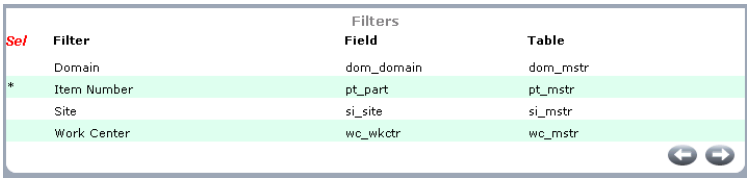
These settings determine whether electronic signature processing occurs for data associated with specified values.

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.

See “Filters” on page 186.

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.

Fig. 13.13
E-Signature Workbench Profile Maint, Filters



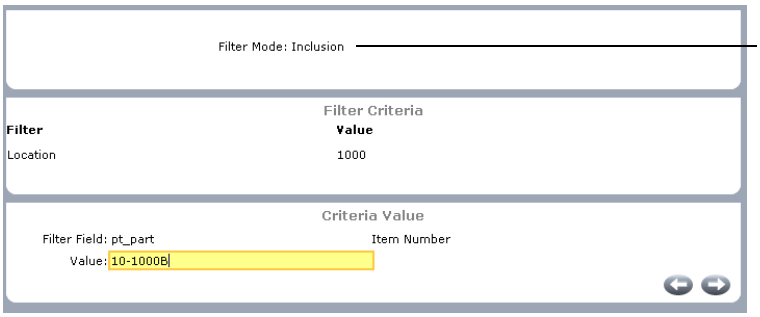
The Filter Criteria frame lists all the filters that were selected in the Filters frame. To enter criteria values for a filter, navigate to the Criteria Value frame and enter a value that will be used to either include or exclude electronic signature processing, depending on the filter mode.

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 exercise caution 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.

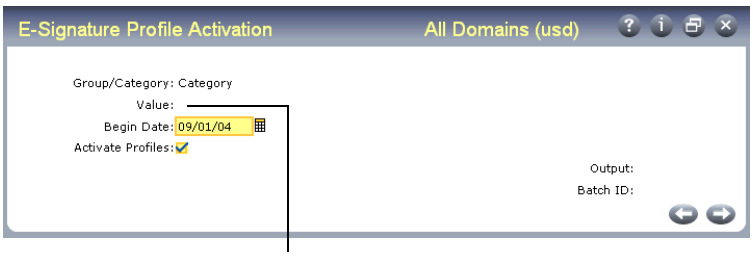
Fig. 13.14
E-Signature Workbench Profile Maint, Filter Criteria and Value



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. 13.15
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 184.

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 members of the administrator group specified in Security Control (36.3.24).

See “E-Mail Notifications” on page 202.

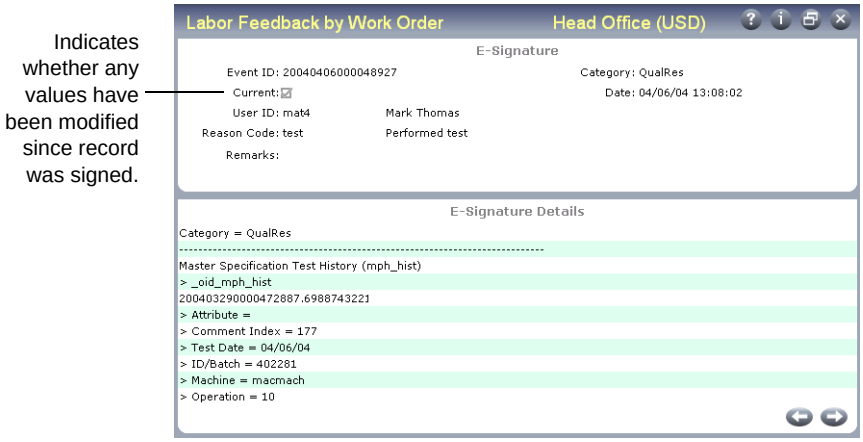
Recording Electronic Signatures

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. For example, Figure 13.16 shows the latest signature for Labor Feedback by Work Order (17.1).

Note Latest signature data for category 0007, Transaction History, is not included in the display for programs associated with that category. See page 183.

Fig. 13.16
Latest Electronic Signature Display



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.

Note 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 184.

Note If the data about to be displayed has never been signed, the system displays a message for the associated category.

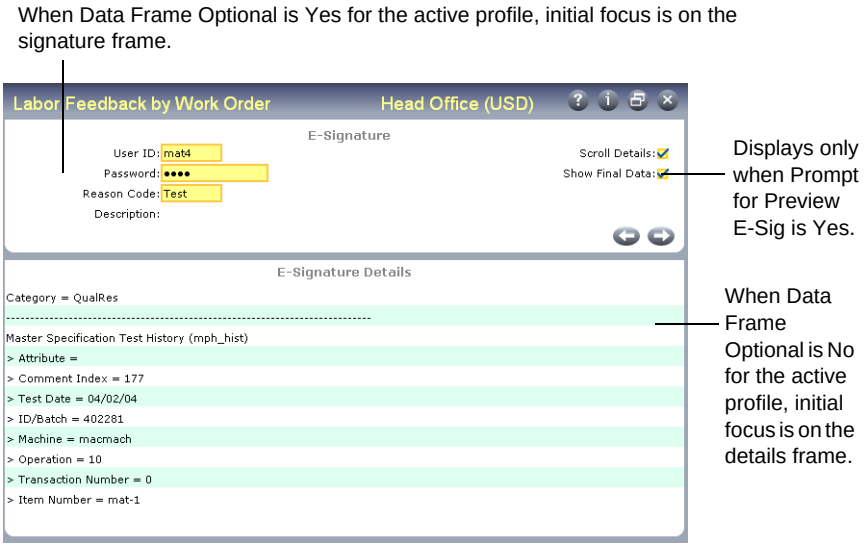
You can scroll through the frame to view all the field values. Choose End to exit from the details frame and return to the program.

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 201.

The prompt screen includes the signature frame, as well as a details frame showing the data being signed. Figure 13.17 shows an example prompt from Labor Feedback by Work Order.

Fig. 13.17
Electronic Signature Prompt



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. 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. Depending on the Prompt for Remarks field in the active profile, you may also be able to enter a remark related to the signature.

See “Prompt for Preview E-Sig” on page 193.

Note The User ID field must be the same as your log-in ID.

If for some reason 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 log-in, 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, deactivate your user ID, and inform the system administrator of a potential unauthorized access attempt. See “Review Security Control Settings” on page 188.

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. Figure 13.18 shows an example.

Fig. 13.18
Signature Completion Message

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.

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 Go, 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 Go 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 Go, and choose End in the second frame, the changes you made in the first frame are not saved to the database. You must choose Go through all the frames to save any changes you make in the program.

See “Apply Profile to Menu Programs” on page 195.

Product Change Control

If you use electronic signatures with the Product Change Control (PCC) module, 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 administrator group set up 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 Max Access Failures value in Security Control.

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 (25.12).

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 process. If you have any questions about
this E-mail, contact the system administrator. Do not reply to this E-mail.
```

See "Review Security Control Settings" on page 188.

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 group identified in User Security Control for
. 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 process. If you have any questions about
this E-mail, contact the 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 the administrator group 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 deactivated 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 group
identified in Security Control for .

User ID deactivated for exceeding max e-sig failures allowed: XXX

This email was automatically generated from a process. If you have any questions about
this email, contact the 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:

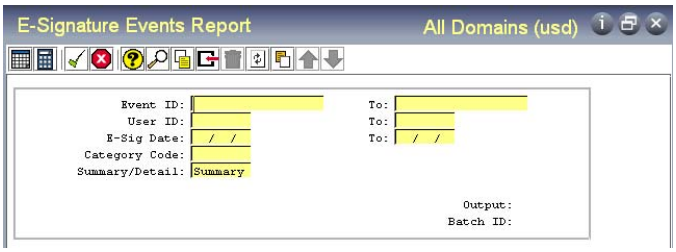
- Display signature events based on information related to the signature itself, such as the user, date, and meaning.
- Select database records based on ranges of values for fields in category top tables, and generate a report on related electronic signature history.
- Monitor log-in history records 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.

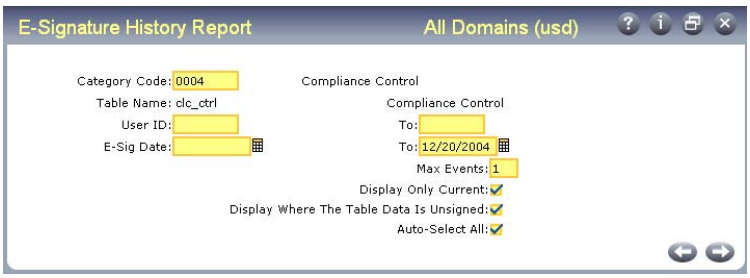
Fig. 13.19
E-Signature Events Report (36.12.4)



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. 13.20
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. 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.

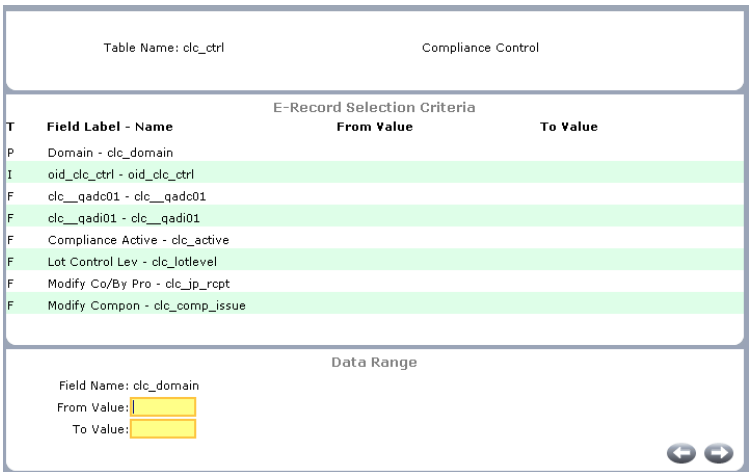
Auto-Select All. Indicate if you want all the fields in the top table to be included in the report by default. You can modify the setting for individual fields as needed in the Report Display Fields frame. The default is Yes. See Figure 13.22 on page 206.

Note Category is a required field.

Press Go 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.

Fig. 13.21
E-Signature History Report, E-Record Selection Criteria



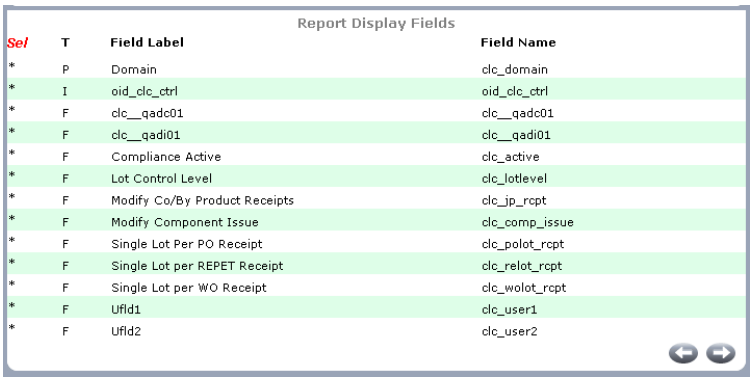
This frame displays the name, label, and type for each field in the top table of the selected category. Field types are Primary (P), Indexed (I), or non-indexed (F). Any selection criteria entered in the Data Range frame display next to the corresponding field on the E-Record Selection Criteria frame. These selection criteria are used to narrow the search results.

See “Top Tables” on page 185.

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 Go. 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.

Fig. 13.22
E-Signature History Report, Report Display Fields


Report Display Fields

Sel	T	Field Label	Field Name
*	P	Domain	clc_domain
*	I	oid_clc_ctrl	oid_clc_ctrl
*	F	clc__qadc01	clc__qadc01
*	F	clc__qadi01	clc__qadi01
*	F	Compliance Active	clc_active
*	F	Lot Control Level	clc_lotlevel
*	F	Modify Co/By Product Receipts	clc_jp_rcpt
*	F	Modify Component Issue	clc_comp_issue
*	F	Single Lot Per PO Receipt	clc_polot_rcpt
*	F	Single Lot per REPET Receipt	clc_relot_rcpt
*	F	Single Lot per WO Receipt	clc_wolot_rcpt
*	F	Ufld1	clc_user1
*	F	Ufld2	clc_user2

All fields are preselected if Auto-Select All is Yes in the first frame. Select or deselect fields as needed. Then press Go 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 log-in 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 deactivated because of excessive failed signature attempts.

When failed log-in 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.

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 191.
- The menu program has Apply selected. See page 195.

Based on those values, the reports and inquiries listed in Table 13.6 can include the Display E-Signature Details field.

Table 13.6
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 Inquiry	3.2	0006
Inventory Detail by Site Inquiry	3.3	0006
Inventory Detail Report	3.6.5	0006
Inventory Detail by Location Report	3.6.6	0006
Inventory Detail Report	3.6.5	0006
Transactions Detail Inquiry	3.21.1	0007
Operation Transaction Detail Inquiry	17.9	0003
Operations by Work Order Report	17.14	0003
Operations By Employee Report	17.15	0003
Quality Order Results Report	19.12	0008
Certificate of Analysis Print	19.20	0008
Control Table 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 and optionally delete electronic signature records from the system when they are no longer needed online.

Note You cannot delete signature records without archiving them.

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.

Fig. 13.23
E-Signature Archive/Delete (36.12.14.22)

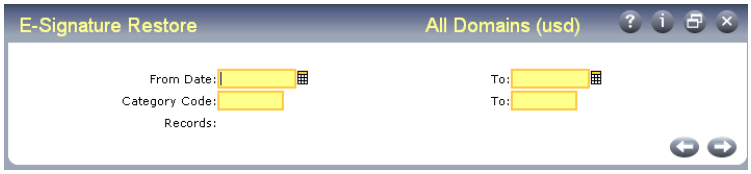
Select records by entering the last electronic signature creation date you want the system to consider. The system selects all records up to that date that have not previously been archived.

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.

Because electronic signatures are typically associated with a high-security environment that emphasizes user accountability, this function is unlike other archive/delete programs. In those programs, records are archived to a data file for storage outside the database. However, E-Signature Archive/Delete instead creates records in the same audit databases used by the audit trail functions of the Enhanced Controls module.

Important So that they can be tracked properly for recovery, records are stored based on the signature event dates. This means that audit databases with connection records in Audit DB Maintenance (36.12.13.11) that have E-Signature set to Yes and that cover the correct date ranges must be running when archiving takes place so that the system can connect to them as needed. The same audit databases must be available when you restore the records. See “Set Up Audit Trails” on page 187.

Fig. 13.24
E-Signature Restore (36.12.14.23)



Chapter 14

Audit Trails

This chapter discusses the following topics:

Overview

Using this feature of the Enhanced Controls module, you can configure your system to maintain audit trails. Audit-trail records are created and stored in an audit database. They contain facts about changes made in the primary database. A typical audit record includes information that helps you identify who made a change, when the change was made, and what the change was. You can set up these functions for all primary database tables or you can limit the audit trail recording activity to specific database tables.

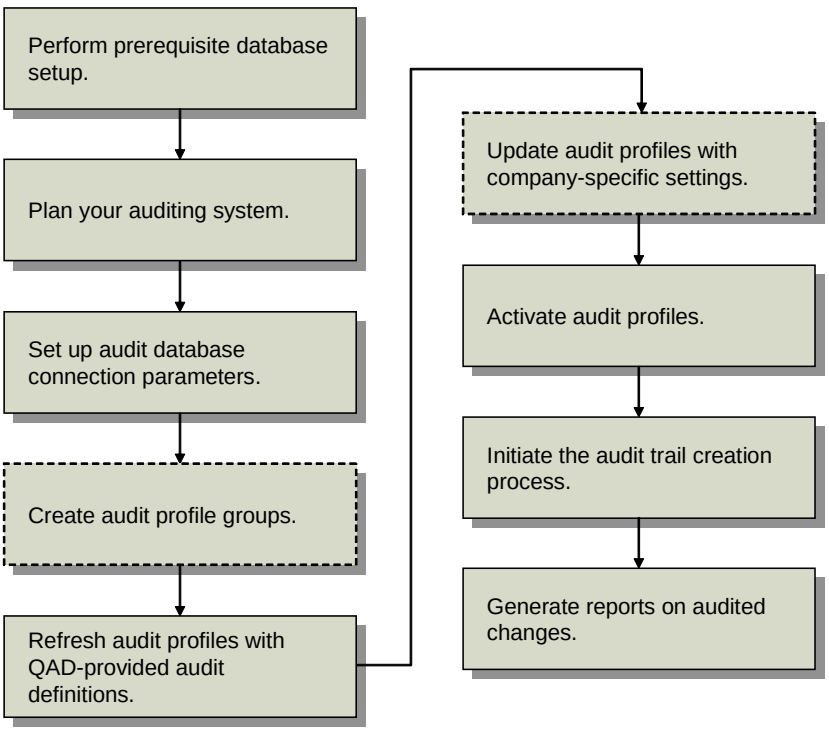
Note Currently only changes to tables in the qaddb database can be tracked.

The auditing system maintains audit information based on an *audit profile*. An audit profile is a definition associated with a specific database table that indicates whether audit functionality is turned on and contains a list of the system and user-defined *delete event keys*. These key fields are used to create search criteria that enable the audit trail reports to find audit information for deleted records.

Auditing Process Work Flow

Use the programs in the Audit Trail Setup Menu (36.12.13) to set up and configure auditing functions. Figure 14.1 illustrates the auditing process work flow. Use it to set up auditing functions in your environment.

Fig. 14.1
Audit Trail Setup Flow



Before the auditing process can begin, the prerequisite implementation and planning steps must be completed. Implementation steps include specifying the unique OID generator code for the database, adding triggers to the Progress database, creating the audit schema holder in Oracle environments, setting up audit databases, and ensuring that a system administrator group has been defined to monitor auditing notifications.

See page 214.

Planning steps include developing a detailed auditing plan containing a list of the tables to be audited and a detailed data management strategy.

See page 216.

Note The audit plan should be part of a detailed security plan to meet your business requirements. See “Security in QAD Enterprise Applications” on page 134.

Within the system, the first activity in setting up auditing functions is to create the records that specify the audit database connection parameters and the effective dates using Audit DB Maintenance (36.12.13.11). You also indicate if each audit database is online or offline. For each connection record, you specify a parameter file or the parameters to use for connecting to the audit database.

See page 217.

Note Electronic signature functionality uses audit databases for archiving signature records when you use E-Signature Archive/Delete (36.12.14.22). You can use the same databases where audit trail information is stored or set up separate audit databases just for archiving signature records. See “Electronic Signatures and Audit Databases” on page 213.

To avoid repetitive data entry for individual table profiles, create audit groups consisting of sets of related tables to audit in Audit Group Maintenance (36.12.13.1), then refresh the table profiles in Audit Workbench Refresh (36.12.13.4) for each group. Table profiles do not exist until they are manually updated with the QAD-provided information using Audit Workbench Refresh.

An audit group is simply a group of tables. Creating an audit group removes the requirement that each table profile must be refreshed individually. When an audit group is refreshed, profiles for all member tables are automatically refreshed. This saves time and can be used to organize table profiles into functionally similar groups.

See page 223.

After refreshing the table profiles, you can manually update profiles in Audit Workbench Profile Maintenance (36.12.13.5) to turn auditing functions on or off and to specify additional delete event keys. Alternatively, the default QAD-provided delete event keys are used if the profile is not updated.

See page 226.

To begin auditing, activate the profiles with Audit Profile Activation (36.12.13.8). Activated profiles are staged for auditing to begin on a future date; auditing does not occur immediately after a profile is activated. On the specified begin date, the system begins generating auditing information for each table profile.

See page 227.

Start the process that commits audit data to the audit database in Audit Trail Creation Process (36.12.13.23). Generated audit information is temporarily staged in a database table where it is retrieved by the audit trail creation process and committed to the audit database. This approach minimizes the impact of generating audit data on system performance.

See page 227.

Use Audit Trail Report – Existing (36.12.1) and Audit Trail Report – Deleted (36.12.2) to report audit information. You can run reports on the audit data only after it has been committed to an audit database and only if the audit database is still online.

See page 231.

Audit Trail Data Flow

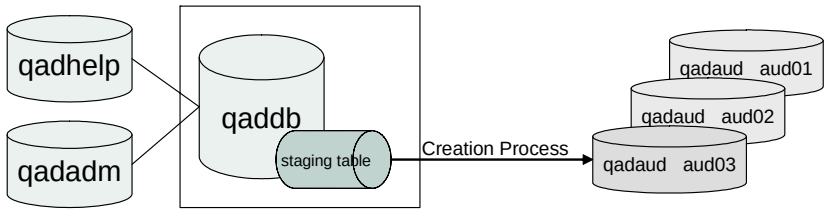
Audit trail functions use a separate database (qadaud) to store the audit trail data. These records are linked to records in the standard database (qaddb) by the unique object ID associated with the records.

Audit data is initially saved in a staging table in qaddb. The audit trail creation process then moves audit data from the staging table and commits it to the audit database. It uses the data defined in Audit DB Maintenance to determine which database to connect to as well as the required connection parameters.

See “Setting Up Database Connections” on page 217.

Figure 14.2 illustrates the basic flow of data.

Fig. 14.2
Audit Trail Creation Process Data Flow



In Figure 14.2, three audit databases are displayed. As part of implementation planning, each company must determine how frequently a new audit database needs to be brought online based on sizing requirements. The size of the audit database depends on the number of tables you decide to audit and the number of changes to records in those tables.

While only one database is updated at a time, you can generate reports for records stored in any number of audit databases.

See the installation guide for guidelines to consider when planning database sizing.

The audit trail creation process can be started automatically by the system administrator using a custom startup script. It can also be started using Audit Trail Creation Process (36.12.13.23). If your environment generates large amounts of audit information, you can run multiple processes.

The creation process runs constantly in a dedicated session, commonly referred to as a background process. It continues to commit data generated prior to 12:00 AM (midnight) until all records for a specific day have been committed to the current audit database. Once it finishes committing data for the day, the system reviews database connection records and connects to a new database if required, based on the database active date setting. It then continues recording activities for the new day.

See “Starting the Audit Process” on page 228.

If the creation process cannot connect to the audit database using the connection records defined for the current day, an e-mail is sent to the system administrators and a message is written in the audit log file. Audit data continues to be stored in the staging table in qaddb, ensuring that no auditable events are missed. Once the audit database becomes available, the Audit Trail Creation Process commits the saved data to it.

See “Audit Trail Creation Process Connection Error” on page 230.

Important System administrators should monitor the log file to ensure the audit update process is running successfully. Certain error conditions do not generate an e-mail message; for example, a server crash.

When delete event data is committed, as an additional safety measure, the system verifies the key field data. In the rare event that the validation fails, the audit data is automatically stored in a backup audit error table. An e-mail is generated notifying the administrator group of the problem. The problem data must be manually corrected by the system administrator. Contact the QAD Support organization for assistance in performing this task.

See “Audit Trail Creation Process Write Error” on page 230.

Electronic Signatures and Audit Databases

If you also use the Electronic Signatures function, you should be aware that there are relationships between that functionality and the audit databases.

The electronic signature archive function does not work the same way as typical archiving, which copies records from the database to operating system files. In electronic signature archiving, the system copies records to an online audit database, from which they can be restored to the system.

See “Archiving and Restoring Records” on page 208.

Depending on database setup, you can either:

- Use the same database that stores audit trail information.
- Create a separate audit database specifically for archived electronic signature records.

See “Setting Up Database Connections” on page 217.

Note Regardless of which method you use, the audit databases that apply to the electronic signature records being archived or restored must be online when you run the archive or restore function.

Completing Prerequisite Activities

Before setting up the auditing features, you must complete the following prerequisite activities:

- Define an OID generator code. The system prompts for this code during installation or conversion.
- Create, configure, and start audit databases. These tasks are performed outside the system using MFG/UTIL before any auditing information can be successfully saved to an audit database. For details, see the installation guide.
- Set up an administrator group in Security Control (36.3.24) to receive automatically generated e-mail notifications related to audit processing. This group also receives e-mail notifications for security violations and failed e-signature attempts. See “Administrator Group” on page 151.

Note These activities, while related to auditing, are described more fully in other documentation. This section provides an overview of the activities and how they affect the auditing process. Refer to the referenced documentation for additional details.

Specify the OID Generator Code

During installation or upgrade conversion, the system prompts you to enter the OID generator code. You can choose any code that you want. However, Progress Software Corporation offers its customers the opportunity to register for unique IDs so other companies will not use the same number, maintaining uniqueness world wide.

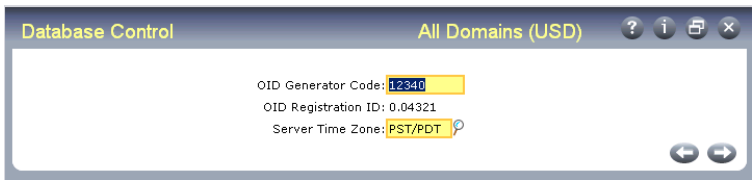
QAD highly recommends that you obtain a code by registering each of your databases for a Dynamics Site Number from Progress Software Corporation. For details and registration, see the information at the following Progress site:

http://www.progress.com/dynamics/sitenumber

In a multiple-database environment, you should obtain an ID for each database. Each qaddb database is assigned its own ID so that the ID values remain unique even when records are merged into one database.

You can use Database Control (36.24) to change the generator code for a particular database. Modifying the OID Generator Code does not change the OID values for existing records. After you make the change, the OID values for all new records will include the new generator code.

Fig. 14.3
Database Control (36.24)



Based on the OID generator code, the OID fields in the database are populated using an algorithm that ensures uniqueness across all records, tables, and databases within the company. The value stored in the OID field for each record has the following decimal format:

<date><seq_value>.<registration_id>

Where:

<date> is the server date with format yyymmdd.

<seq_value> is obtained from a Progress database sequence.

<registration_id> identifies the origin of the OID value.

The registration ID is derived from the OID generator code by reversing the digits of the generator code value and placing the decimal point in front of the result.

Create and Configure Audit Databases

Audit databases must be created outside of the system. This guide assumes you have already created one or more audit databases. You must have an audit database configured and running before the auditing functions can connect to it. The audit database creation and maintenance tasks are detailed in the installation guide.

You define the parameters for connecting to the auditing database using Audit DB Maintenance. This activity is described in this chapter.

See “Setting Up Database Connections” on page 217.

Note Electronic Signatures also uses audit databases for storing archived records. As part of the creation and configuration process, you should consider whether you want to use separate audit databases just for this purpose, or use the same database for both audit trails and electronic signature archives.

See “Identifying the Database Type” on page 219.

Define an Administrator Group

Audit trail functions use the administrator group and e-mail functions to send e-mail alerts when profiles are activated, audit information cannot be committed to an audit database, or a connection to the audit database cannot be created successfully.

If an administrator group is not already defined in Security Control (36.3.24), specify an existing user group as the administrator group. Alternatively, create a new user group in User Group Maintenance (36.3.4) and define it as the administrator group.

See “Administrator Group” on page 151.

Each member of the group must have an associated e-mail definition specified in the user profile record created in User Maintenance (36.3.1) in order to receive e-mail notifications from the system.

See “E-Mail Notifications” on page 229.

Planning an Auditing System

Every environment has unique record-keeping requirements. Before you begin setting up the auditing functions, consider creating:

- A detailed data retention plan including details such as:
  - A detailed list of the types of information you need to audit
  - A detailed list of the database tables that contain information you need to audit
- A system resource and hardware plan with disk space and system resource availability
- A maintenance schedule for planning when new audit databases are created and brought online
- An information retention plan detailing how long auditing information is kept online for reporting purposes
- An archive plan detailing when audit databases are taken offline and where they are stored

Consider the following points:

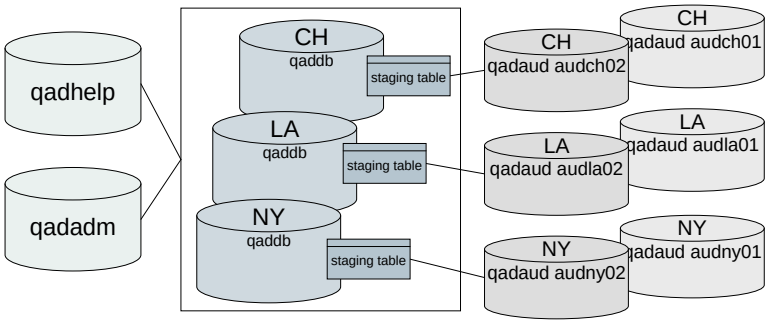
- The number of database tables you maintain information for and the number of audited transactions occurring on your system are directly related to the system resource usage and availability. These factors directly affect maintenance requirements and administrative overhead.
- If you plan to use Electronic Signatures functionality, the frequency with which you will archive signature records can have a significant impact on whether you set up signature-specific databases or simply use the same databases for both audit trails and signature archives.
- The information that is important to you may reside across multiple database tables. Additional research is required to find all the database tables containing this information.
- Depending on which modules you are licensed to use, some tables in the database are not used and can be disregarded.
- For research purposes, you should be familiar with the Progress Database Dictionary and the *Entity Diagrams* and *Database Definitions* reference guides.
- Some environments, such as in the medical industry, have very specific and stringent electronic information retention regulations; make sure you are familiar with any regulations or standards in your specific industry or region.
- For additional security, you may want to store offline audit databases on read-only media such as CD-ROMs.

Multi-Database Environments

Implementing auditing in multi-database environments has no additional special requirements. Each database in these environments has its own audit database. You set up and define audit profiles in each database, then use different audit databases to store the audit information. Each database requires separate connection records for its specific audit databases. An audit trail creation process runs in each database, committing audit information to the appropriate audit database.

Figure 14.4 shows a typical multi-database environment layout. Notice that each database has its own unique audit staging table and audit databases.

Fig. 14.4
Multi-Database Environment Layout



Setting Up Database Connections

Use Audit DB Maintenance (36.12.13.11) to create and maintain connection parameters and active date range information for the system audit databases. These connection parameters are used by these functions:

- Audit Trail Report–Existing and Audit Trail Report–Deleted use the connection records to connect to the proper audit databases when reporting audit information.
- The Audit Trail Creation Process uses the connection record information to connect to the current active audit database when it needs to commit audit information from the staging area.
- If you use the Electronic Signatures function, the system uses the records to connect to the appropriate databases during signature archive and restore activities. See “Archiving and Restoring Records” on page 208.

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.
- You must specify a type of ORACLE if you are connecting to an Oracle database through a Progress schema holder.
- Connections to the audit databases are not permanent. The audit reports and Audit Trail Creation Process—as well as electronic signature archive and restore functions—use the connection information to connect to the audit databases as needed. These processes do not maintain a connection to an audit database after they have retrieved or committed the information they are handling.

- The system can connect to multiple audit databases simultaneously.

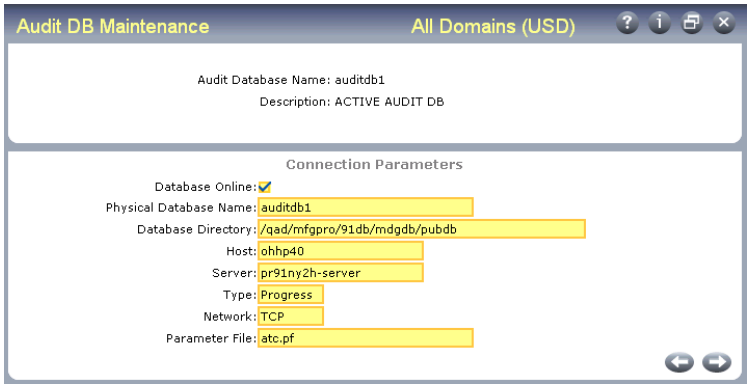
Important Audit databases must be configured and running before connecting to them using the connection parameters. Audit DB Maintenance does not start or stop audit databases. It only stores the connection parameters used to connect to them. You must set up external procedures to start up and shut down audit databases as needed.

Example When bringing up the system after a scheduled shutdown, a script is executed from the operating system. The script can be created from MFG/UTIL. Using a predefined list, the script starts up the audit databases for reporting. When the audit reports are run, they use the connection record parameters to connect to the appropriate databases and report audit information as required by the report selection criteria.

Database connection parameters are defined by the way audit 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.

For details, see the installation guide.

Fig. 14.5
Audit DB Maintenance (36.12.13.11)



In the first frame, enter a name for this audit database connection record. The name must be 8 or less 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 audit database.

Specifying Database Connection Parameters

For Progress databases, you can specify connection parameters directly in Audit DB Maintenance or include them in a parameter file. If you use a parameter file, you must still specify the database name in the Audit DB Maintenance screen. If the database is not located in the PROPATH, you must also specify the full path in Database Directory.

See “Using a Parameter File” on page 220.

Note Typically, each implementation has additional parameters that are needed to accommodate specific requirements that cannot be specified directly in the maintenance screen. These can be supplied in the parameter file.

For Oracle databases, you specify some values in Audit DB Maintenance, but a parameter file is always required. This is because two sets of connections are required for Oracle databases: one for the Progress schema holder and one for the Oracle database. The values that you specify in the Audit DB Maintenance fields apply to the Progress schema holder only. Connection to the Oracle database is defined in the parameter file.

Database Online. This field indicates whether the system should attempt to connect to the audit database. It does not indicate that an audit database is running, nor 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 or schema holder. 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.

For example, on a UNIX system if your database is stored as `/qad/d7/qadaud01.db`, then the directory is set to `/qad/d7` and the physical name is `qadaud01`. You do not need the `.db` extension.

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.

The following fields should only be specified if they are not included in a parameter file. If you do specify them with a parameter file, any values in the parameter file are used instead of these values.

Host. Enter the name of the host server where the Progress database or schema holder 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.

Type. Specify the audit database type, either Progress or Oracle. You must use a parameter file to connect to Oracle audit databases.

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.

See “Using a Parameter File” on page 220.

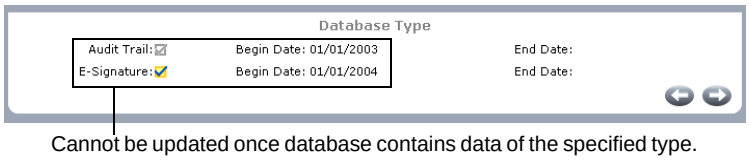
Identifying the Database Type

The audit database can serve two functions:

- Store audit trail records
- Store archived electronic signature records

Depending on the values you enter in the Database Type frame, an audit database can serve either or both of these purposes.

Fig. 14.6
Audit DB Maintenance, Database Type Frame



Audit Trail, E-Signature. If you use both Audit Trails and Electronic Signatures functions, use these fields to control whether audit trail data and archived signature records are stored in separate databases. When both fields are Yes, the system uses this database for both data types. Otherwise, you can choose to set up and administer a separate database for each function.

Begin Date. Enter the date when the system should begin saving the specified type of information to the audit database identified by this connection record. Connection records become effective at 12:00 AM on the indicated date.

Setting Database Online to Yes does not start an audit database; it simply indicates that the system is allowed to save or retrieve information for the audit database. This assumes that the database has already been configured and started outside of the system. The Begin Date must correspond to a date when the indicated database is already online; otherwise, the system reports connection errors. New audit trail creation processes cannot be launched if the system cannot connect to the current audit database. This is also true of electronic signature archive and restore activities.

End Date. This value cannot be updated manually. It is automatically updated when the system begins storing information of the specified type in a new audit database.

Note You cannot change a data type or begin date field once the system has used this database to store any data of the associated type.

Using a Parameter File

Depending on your environment, your parameter files may differ from these general guidelines.

The following guidelines apply to parameter files used to connect to a Progress database or Progress schema holder for an Oracle database:

- Specify the parameter file name in Parameter File.
- Specify connection parameters in the parameter file or in the corresponding fields. Do not include the parameters in both the file and in the corresponding fields.
- The parameter file must be accessible through the PROPATH or located in the directory specified in Database Directory.
- If you use a parameter file, you must still specify the database name in the Audit DB Maintenance screen. If the database is not located in the PROPATH, you must also specify the full path in Database Directory.
- The parameter file should not include either the `-ld` or `-db` parameters.
- The parameter file must include the `-trig` parameter that specifies the location of trigger files.

For Oracle audit databases, you specify the schema holder name in Physical Database Name and set Type to Oracle. The parameter file must include the parameters for the schema holder followed by parameters for connecting to the Oracle database, in this order:

- `-trig` parameter
- Other connection parameters for the schema holder
- `-db` parameter for the Oracle database
- The Oracle database connection parameters, including the `-dt ORACLE` parameter and the `-ld` parameter, if needed

Table 14.1 shows several sample database connection settings and corresponding parameter files.

Table 14.1
Sample Parameter Files and Audit DB Maintenance Settings

Database Type	Audit DB Maintenance Settings	Parameter File Values
Progress	Physical Database Name: auditdb1 Database Directory: /qad/mfgpro/db Host: mainserver Server: audit1-service Type: Progress Network: TCP Parameter File: example1.pf	`-trig “triggers”`

Progress	Physical Database Name: auditdb1
	Database Directory: /qad/mfgpro/db
	Host:
	Server:
	Type: Progress
	Network: TCP
	Parameter File: example2.pf
		`-trig “triggers”`
		`-H mainserver`
		`-S audit1-service`

Database Type	Audit DB Maintenance Settings	Parameter File Values

Oracle	Physical Database Name: audit
	Database Directory: /qad/mfgpro/db
	Host: mainserver
	Server: otest3-service
	Type: ORACLE
	Network: TCP
	Parameter File: example3.pf
		`-trig “triggers” -znotrim`
		`-db otest3`
		`-dt ORACLE`
		`-U qad@otest3`
		`-P QAD`

Oracle	Physical Database Name: audit
	Database Directory: /qad/mfgpro/db
	Host:
	Server:
	Type:
	Network: TCP
	Parameter File: example4.pf
		`-trig “triggers” -znotrim`
		`-H mainserver`
		`-S audit1sh-service`
		`-db otest3`
		`-dt ORACLE`
		`-U qad@otest3`
		`-P QAD`

Setting Up Audit Profiles

Setting up and using audit profiles include these steps:

- Create audit groups
- Refresh audit profiles
- Update audit profiles
- Activate profiles

Overview

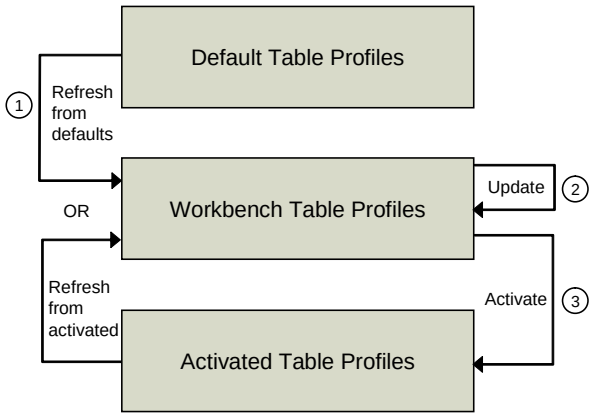
Each database table has its own profile. Initially all table profiles are empty; they must be refreshed with the QAD-provided default information. Table profiles hold values that auditing functions use to manage the audit trail generation and reporting process. This information affects auditing only after the profile is activated.

A table profile:

- Indicates whether auditing is enabled
- Maintains a list of QAD-defined delete event keys
- Maintains a list of user-defined delete event keys

In addition to the QAD-provided default data, the system maintains two sets of profiles: the profiles you edit in the workbench and the activated profiles. 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 Audit Workbench Profile Maintenance without affecting the active system.

Fig. 14.7
Table Profiles



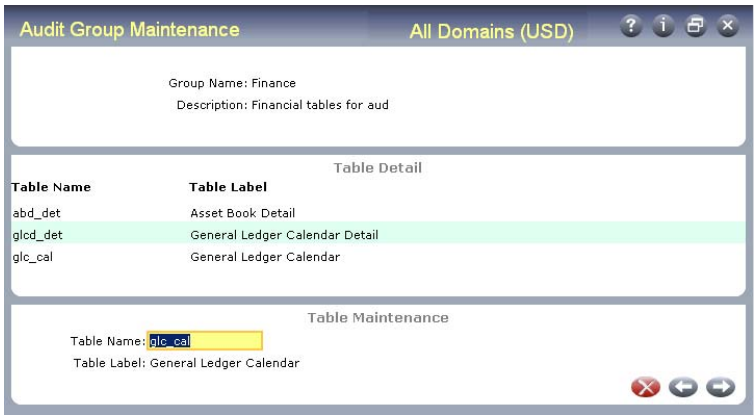
Before refreshing workbench profiles, you can optionally create audit groups to manage several profiles more easily and streamline the data setup process. Once refreshed, modify the profiles with your requirements. You can enable or disable auditing and add delete event keys as needed. When your profiles are complete, activate them and set a begin date. To discontinue auditing a table, simply update the workbench profile to set Audit Trail to No; then activate it with the begin date set to the date auditing should stop.

Creating Audit Groups

Use Audit Group Maintenance (36.12.13.1) to group all the tables you plan to audit, or to group related database tables for auditing purposes. Audit groups streamline the setup process by letting you refresh and activate the profiles for all the member tables at once, instead of maintaining one table profile at a time.

For example, set up a financial audit group to track financial transactions and a manufacturing group to separately maintain the profiles for manufacturing transactions. Add tables to each group to accommodate your specific data tracking and maintenance requirements.

Fig. 14.8
Audit Group Maintenance (36.12.13.1)



Specify a group name, up to 8 characters. An audit group cannot have the same name as a database table. Then provide a brief description and press Go to display the Table Maintenance frame. In this frame, add as many tables to the group as required. After they have been added, the tables display in the Table Detail frame.

Use Audit Group Report (36.12.13.2) to display the records defined in this program.

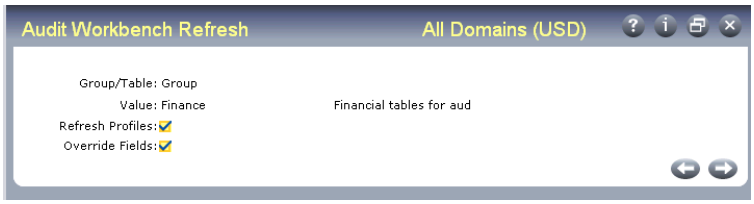
Refreshing Profiles

When initially setting up auditing functions, workbench table profiles are empty, and must be manually populated. Use Audit Workbench Refresh (36.12.13.4) to update the empty profiles with the QAD-provided default information. You can refresh one table at a time or, optionally, refresh the profiles for an entire group of tables.

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

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

Fig. 14.9
Audit Workbench Refresh (36.12.13.4)



Indicate if you want to refresh tables or groups; then specify the table name or group name to be refreshed. Leave the Value field blank to refresh all tables or groups, based on the setting in the Group/Table field.

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.

No: The Refresh Profiles frame does not display.

Yes: The Refresh Profiles frame displays. Use it to specify the source profiles for the refresh. You can use active profiles or the QAD-provided default profiles.

Override Fields. Indicate whether to override the field that controls auditing for the specified profiles. The QAD-provided default profiles have auditing functions turned on.

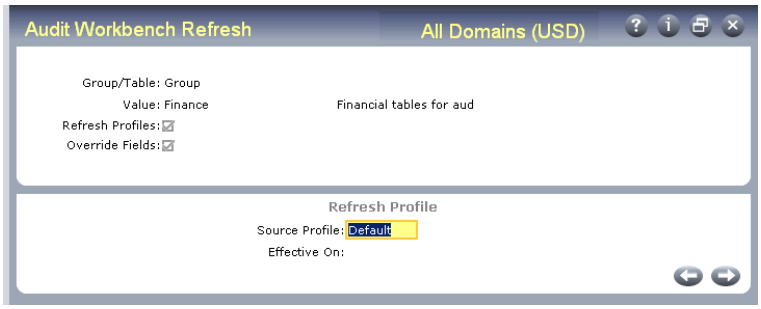
No: The Override Fields frame does not display.

Yes: The Override Fields frame displays for you to set the value for Audit Trail to Yes or No for all the specified profiles. If Refresh Profiles is Yes, the value specified here replaces the refreshed value.

Refresh Profile Frame

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

Fig. 14.10
Audit Workbench Refresh, Refresh Profile Frame



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

Active: Each specified workbench profile is refreshed using the activated profiles in use on the date specified in Effective On. The corresponding table 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 audit functions to load the QAD-provided values into the profiles for the tables you plan to audit.

Effective On. 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 effective date. If an activated profile was not in use on the date, an error displays and the target profile is not refreshed.

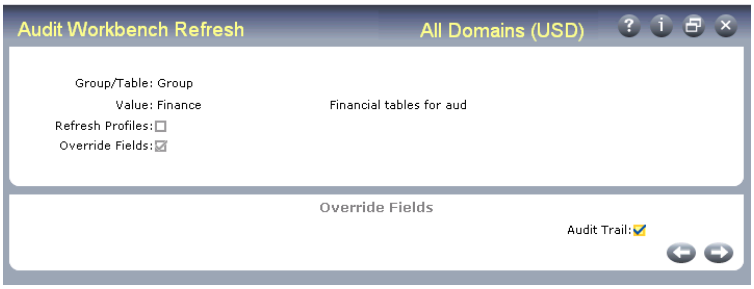
Note This field is available only when Source Profile is Active.

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.

Fig. 14.11
Audit Workbench Refresh, Override Fields Frame



Audit Trail. Indicate whether to enable auditing for the tables being refreshed.

If Refresh Profiles is No, the value specified here replaces the Audit Trail value in the current workbench profiles for the specified group or table.

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

Use Audit Workbench Profile Maintenance to change this value for individual table profiles.

Updating Audit Profiles

Use Audit Workbench Profile Maintenance (36.12.13.5) to adjust profile settings for your specific environment. You can enable or disable auditing and add user-defined delete event keys for the tables you plan to audit.

To disable auditing for a table already being audited, you must create a new activated profile for that table. Do this by updating the workbench profile with Audit Trail set to No; then activate that profile with the proper begin date.

Delete Event Keys

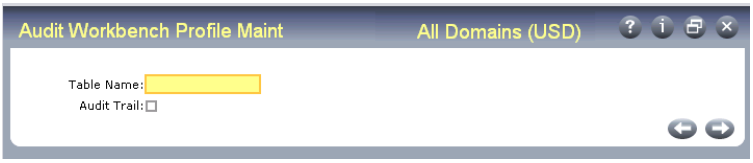
The primary index fields for each database table are defined as delete event keys. These keys are stored in the audit database when a record in this table is deleted. This is in addition to the standard information stored for create and modify auditing events. The system uses this data to uniquely identify a deleted record. This search criteria is used by Audit Trail Report–Deleted to find and retrieve deleted record information from audit databases.

Example To search for and report audit trail information for a deleted sales order, Audit Trail Report–Deleted uses a date range and the so_nbr field values indexed when the record was originally deleted. The so_nbr field is the QAD-defined delete event key for the so_mstr table.

Updating Profiles

Figure 14.12 illustrates the first frame of Audit Workbench Profile Maintenance.

Fig. 14.12
Audit Workbench Profile Maint (36.12.13.5)



Enter a table name and press Go. Then indicate if you want audit trail to be enabled or not. Press Go to display the delete event keys currently defined for the table.

Fig. 14.13
Audit Workbench Profile Maint, Delete event Key Detail

In addition to the QAD-defined delete event keys, you can manually define any other field in the table as a delete event key in the Delete Event Key Maintenance frame.

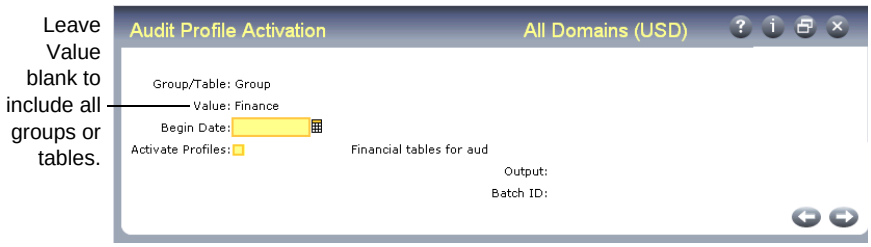
Note You cannot delete the QAD-provided keys.

Use Audit Workbench Profile Report (36.12.13.6) to display the information updated in this program.

Activating Audit Profiles

After completing the workbench table profiles, use Audit Profile Activation (36.12.13.8) to activate profiles for one table or a group of tables. Activated profiles are staged for auditing to begin on a future date; auditing does not occur immediately after a profile is activated. On the specified begin date, the system begins generating auditing information as defined by each activated table profile.

Fig. 14.14
Audit Profile Activation (36.12.13.8)



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 begin date.

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

When this program completes execution, it generates a report that displays information for each table in the activated profile that includes:

- The table name and description

- Information from the active profile being replaced, if one existed, including the previous setting for begin date, audit trail, and the delete event keys
- Information from the now active profile, including the new setting for begin date, audit trail, and the delete event keys

If Activate Profiles is set to No, only the report is generated; the profiles currently in use are not updated.

When Activate Profiles is Yes and all processing is complete, an e-mail is sent to the system administrator group defined in Security Control notifying them of the changes to the activated audit profiles.

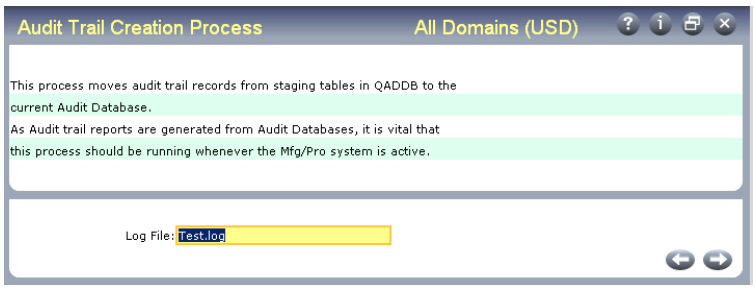
Use Activated Audit Profile Report (36.12.13.9) to display details about activated profiles.

See “Audit Profile Activation E-Mail” on page 229.

Starting the Audit Process

Use Audit Trail Creation Process (36.12.13.23) to start a background process that moves staged audit data from the staging table in qaddb to the appropriate audit database.

Fig. 14.15
Audit Trail Creation Process (36.12.13.23)



To commit generated audit information to the audit database, the Audit Trail Creation Process should be running when any active audit profiles are scheduled to become effective.

Consider using a CIM startup script to start this process automatically after system backups or downtime. Otherwise, it must be manually started whenever the system is restarted.

Note The installation guide includes a sample CIM file and startup scripts for executing this program.

Audit Trail Creation Process generates e-mail messages to the system administrator group when connection errors occur. The system also saves time-stamped messages related to database connection to the log file specified in this program. These messages record when an audit process starts and ends, as well as any connection errors. Messages look like the following samples:

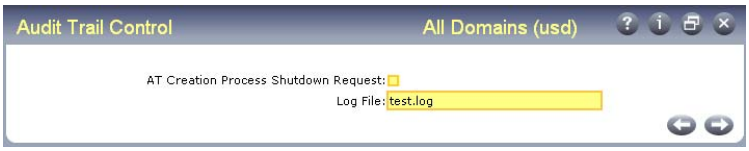
```
2003-12-05 @ 14:26:53 AT Creation Process session begin
2003-12-05 @ 14:27:43 Database connection failed 12-04-03
2003-12-05 @ 14:27:43 AT Creation Process session end
2003-12-05 @ 14:28:11 AT Creation Process session begin
2003-12-05 @ 14:28:11 This process was shutdown from User Account Ctrl
2003-12-05 @ 14:28:11 AT Creation Process session end
```

The log file name defaults from the value specified in User Accountability Control (36.12.13.24). You can specify a full path and file name for the log file; if only the file name is specified, the file is located in the directory where the session running the audit trail creation process was started.

If multiple processes are being used, you can specify a different log file for each. If you do not specify separate log files and multiple sessions are started from the same directory, messages from each process are saved to the same log file.

See “Audit Trail Creation Process Connection Error” on page 230.

Fig. 14.16
Audit Trail Control (36.12.13.24)



To start up the creation process, the AT Creation Process Shutdown Request field must be No in Audit Trail Control.

Shut down the Audit Trail Creation Process by exiting the session from where it was started. Alternatively, use Audit Trail Control to shut down all Audit Trail Creation Processes by setting the shutdown field to Yes.

E-Mail Notifications

The system generates and sends e-mails to the administrator group specified in Security Control (36.3.24) in the following situations:

- One or more audit profiles are activated.
- Errors occur when Audit Trail Creation Process writes to the audit database.
- Errors occur when Audit Trail Creation Process connects to the audit database.

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 (1.12).

The auditing messages all have a comment type of AT. 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 with a reference of `email_postfix` is appended. This comment contains the following information that applies to all system-generated auditing e-mails:

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

Audit Profile Activation E-Mail

Comment Reference: email_profile_activation

Comment Type: AT

The e-mail sent for audit profile activation is similar to this example.

```
The purpose of this E-mail is to inform you that one or more audit trail workbench
```

```
profiles has been activated. You have been included in this E-mail distribution because
you belong to the Administrator group identified in User Security Control. The
information listed below regarding the activation can be used to obtain a detailed report
of the activation by running the Audit Activated Profile Report.

The activation was performed by User ID:

The newly activated profiles are set to begin on date: dd/mm/yy

The number of newly activated profiles with the audit trail enabled:

The number of newly activated profiles with the audit trail disabled:

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

Audit Trail Creation Process Write Error

Comment Reference: email_audit_creation

Comment Type: AT

The e-mail sent when the Audit Trail Creation Process detects an error while writing to the audit database is similar to this example.

```
The purpose of this E-mail is to inform you that one or more processing errors occurred
during Audit Creation Process. You have been included in this E-mail distribution because
you belong to the Administrator group identified in User Security Control.

Total Number of Processing Errors: #

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

When delete event data is committed, as an additional safety measure, the system verifies the key field data. In the rare event that the validation fails, the audit data is automatically stored in a backup audit error table. An e-mail is generated notifying the administrator group of the problem. The problem data must be manually corrected by the system administrator. Contact the QAD Support organization for assistance in performing this task.

Audit Trail Creation Process Connection Error

Comment Reference: email_creation_prefix

Comment Type: AT

The e-mail sent when Audit Trail Creation Process detects an error while connecting to the audit database is similar to this example.

```
The purpose of this E-mail is to inform you that the Audit Trail Creation Process was
terminated. This termination is an abnormal event and should be regarded seriously. You
have been included in this E-mail distribution because you belong to the Administrator
group identified in User Security Control.

<Specific connection error>

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

Three specific error messages can display in the e-mail:

- Audit database is not online. This error displays when the Database Online field is set to No in Audit DB Maintenance.

- Database connection failed. This error displays when the Database Online field is Yes for the audit database in Audit DB Maintenance (36.12.13.11), but the connect statement failed. See “Database Online” on page 219.
- Parameter file not found. This error displays when the parameter file specified for the database in Audit DB Maintenance cannot be found. See “Using a Parameter File” on page 220.

Reporting Audit Data

Use Audit Trail Report–Existing (36.12.1) and Audit Trail Report– Deleted (36.12.2) to review the historical audit trail information maintained in the online audit databases. Both reports function similarly, but one displays audit information for existing records, while the other displays auditing information for deleted records.

These reports use the connection records maintained in Audit DB Maintenance to connect to the audit databases. They connect only during the report generation process; connections to the audit databases are not permanent.

See “Setting Up Database Connections” on page 217.

Important Audit databases must be configured and running before running either report. The report programs do not start or stop audit databases. You must set up external procedures to start and shut down audit databases as needed. If the databases required by the report dates are not available, error messages are generated.

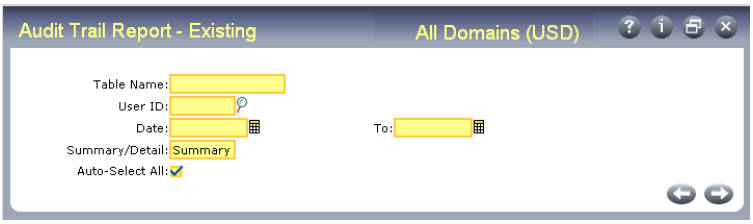
Displaying Existing Audit Data

Use Audit Trail Report–Existing (36.12.1) to find audit information related to existing database records. You can only report against audit databases that are currently online.

Note This report does not display audit information for deleted records. To see audit information for deleted records, use Audit Trail Report– Deleted. See “Displaying Deleted Audit Data” on page 233.

Figure 14.17 illustrates the first frame of Audit Trail Report–Existing.

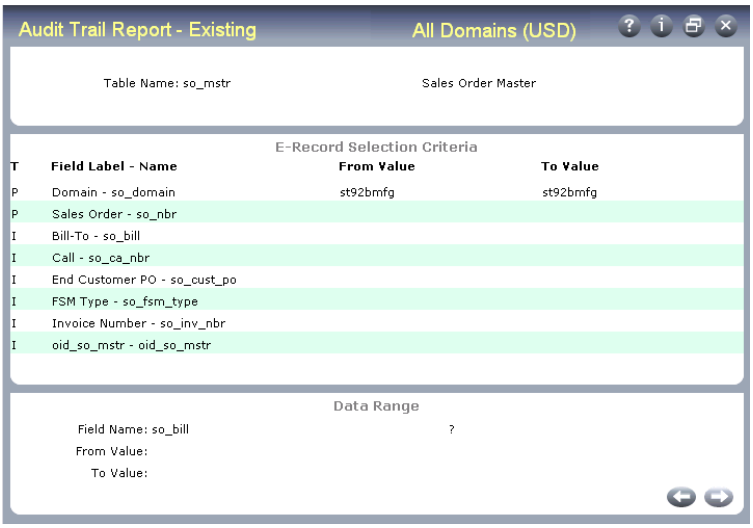
Fig. 14.17
Audit Trail
Report–Existing (36.12.1)



Select the table, user ID, date range, and report style in the first selection criteria frame. You also indicate if you want all the fields in the selected tables to be included in the report by default. You can modify the setting for individual fields as needed in the Report Display Fields frame.

Then press Go to display the E-Record Selection Criteria frame where you can specify a range of values for one or more fields for identifying the records to report.

Fig. 14.18
E-Signature History Report, E-Record Selection Criteria



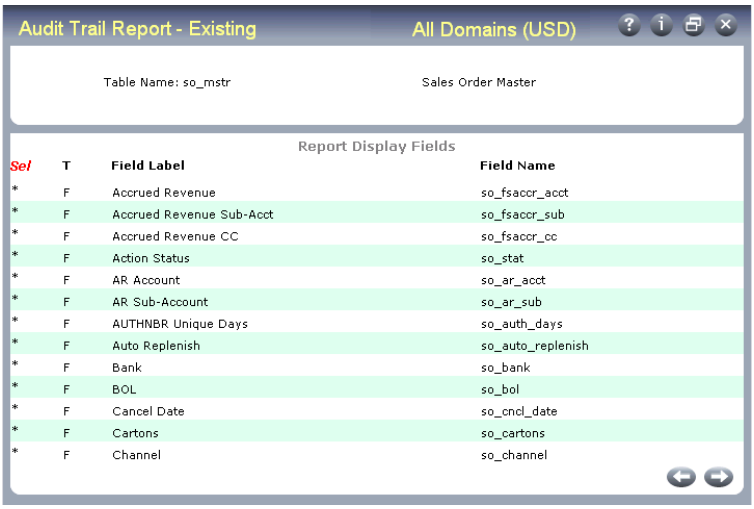
This frame displays the field name, field label, and field type for each field in the selected database table. Field types are Primary (P), Indexed (I), or non-indexed (F). To enter a selection range for a field, navigate to the Data Range frame, where you can specify from and to values. Any selection 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. Not entering a data range for a field matches all values.

Note Large reports may result if you do not specify field-level selection criteria.

To minimize the report output, enter criteria for as many table fields as needed. For example, if you are reporting the audit trails for one or more so_mstr records, scroll to the so_nbr field and press Go. Enter a range of sales order numbers in the so_nbr From Value and To Value fields to narrow the search results. After entering the field-specific selection criteria for your report, press End to continue.

Use the Report Display Fields frame to select or deselect the fields to include or exclude on the resulting report.

Fig. 14.19
Audit Trail
Report–Existing, Report Display Fields



All fields are preselected if Auto-Select All is Yes in the first frame. Select or deselect fields as needed. Then press Go to specify the output device for the report.

Displaying Deleted Audit Data

Audit Trail Report–Deleted (36.12.2) shows a complete history for any audited record that has been deleted. The report includes two date ranges. The first date range indicates the range of audit trail event dates to include in the report. The second date range, Delete Date and To, refers to the date the record was actually deleted.

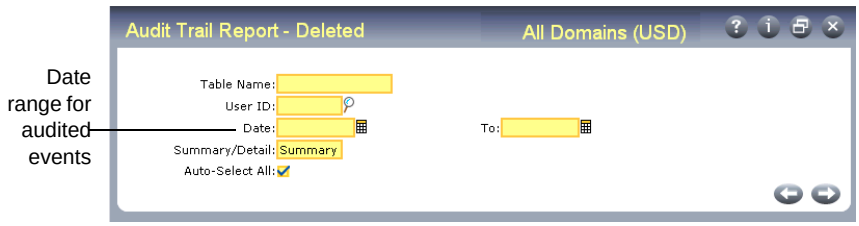
The date range is not mandatory, but entering one narrows the search of audit databases and improves reporting speed. If older audit databases have been take off-line, restrict the date range to dates included in the online databases to avoid error messages.

Example Several records were deleted sometime in June 2007. You need to see the previous year's audit trail information for those records. Enter 07/1/06 to 06/30/07 in the first date range to see events that occurred in the previous year. In the second date range, enter the date range when you think the records were deleted, 06/01/07 to 06/30/07.

The resulting report shows that the last audit event for the reported records was DELETE with an event date of 06/02/07. The report also shows the CREATE and MODIFY events for the records if these events occurred in the previous year.

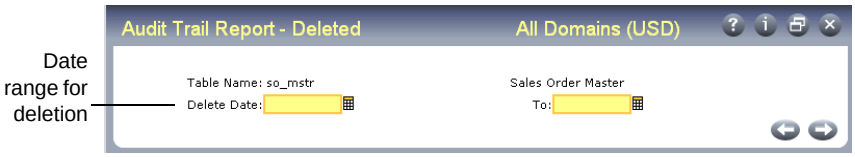
Figure 14.17 illustrates the first frame of Audit Trail Report–Deleted.

Fig. 14.20
Audit Trail
Report–Deleted (36.12.2)



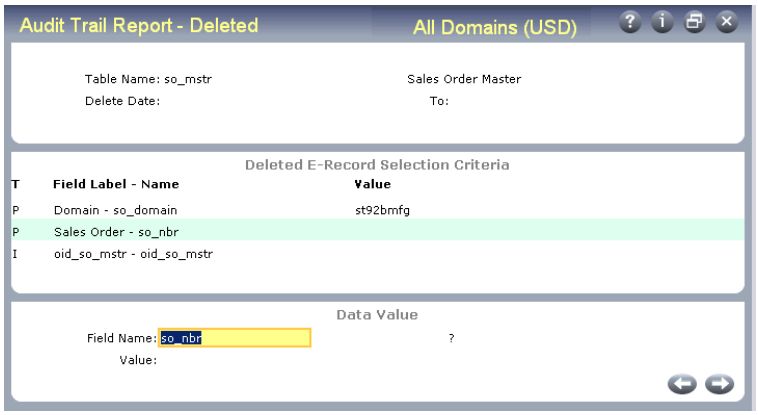
This frame is the same as Audit Trail Report–Existing. Select the table, user ID, date range, report style, and selection default. When you press Go, you are prompted for a date range when the deletions occurred. If you do not know this, enter a date range spanning the online audit databases.

Fig. 14.21
Audit Trail
Report–Deleted, Date Range



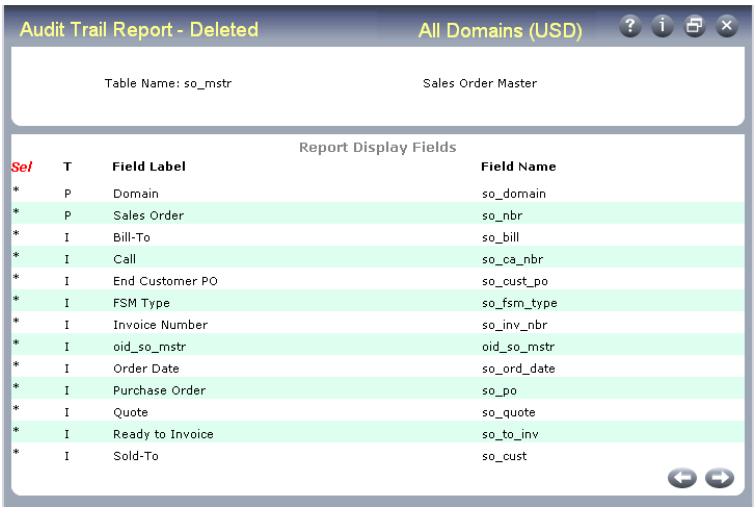
Enter an appropriate date range and press Go. The Deleted E-Record Selection Criteria frame displays.

Fig. 14.22
Audit Trail
Report–Deleted, Deleted E-Record Selection Criteria



The field name, field label, type, and value for the associated delete event keys display in the Deleted E-Record Selection Criteria frame. Field types are Primary (P), Indexed (I), or non-indexed (F). To identify the deleted record, you can enter specific search criteria for each delete event key or choose Go to continue. To enter specific data values for an event key, select the key and then enter the criteria in the Value field. Press End when you finish entering values. The Report Display Fields frame displays.

Fig. 14.23
Audit Trail
Report–Deleted, Report Display Fields



This is identical to the frame that displays in Audit Trail Report–Existing. It displays all the fields in the table for which you are generating a report. All fields are preselected if Auto-Select All is Yes. Select or deselect fields as needed. Then press Go to specify the output device for the report.

Chapter 15

Domain Reference

This chapter includes reference information related to domain changes.

Non-Domain Database Tables

Table 15.1 lists the tables in the database that do not include domain information. These tables contain data that is shared among all domains in a database.

Table 15.1
Non-Domained Tables

Table	Description
abd_det	Asset Book Detail
accd_det	Asset Cost Change Detail
ast_mstr	Asset Master
atak_det	Audit Trail Activated Key Detail
atap_mstr	Audit Trail Activated Profile Master
atc_ctrl	Audit Trail Control
atdc_mstr	Audit Trail Database Connection Master
aterr_mstr	Audit Trail Error Master
atgt_ref	Audit Trail Group – Table Cross-Reference
atg_mstr	Audit Trail Group Master
attmp_mstr	Audit Trail Temporary Master
att_mstr	Audit Trail Table Master
atwk_det	Audit Trail Workbench Key Detail
atwp_mstr	Audit Trail Workbench Profile Master
bkfm_mstr	Bank Account Format Master
ccd1_det	Cost Center/Account Validation Detail
ccd2_det	Cost Center/Sub-Account Validation Detail
cls_mstr	Class Master
cst_mstr	Fixed Asset Custodian Master
ctry_mstr	Country Master
cu_mstr	Currency Master
dbk_mstr	Depreciation Book Master
dbs_mstr	Database Revision Control Master
dc_mstr	Database Connection Master
dmw_wkfl	Draft Management Work Table
dom_mstr	Domain Master
dpc_mstr	Depreciation Convention Master
dpr_mstr	Depreciation Method Master
dprd_det	Depreciation Detail
dprt_det	Depreciation Method Detail
em_mstr	E-mail Master
esapfc_det	E-Signature Activated Profile Filter Criteria
esapfil_det	E-Signature Activated Profile Filter
esapfs_det	E-Signature Activated Profile Filter Set
esapf_det	E-Signature Activated Profile Field

Table 15.1 — *Non-Domained Tables* — (Page 1 of 3)

Table	Description
esapmp_ref	E-Signature Activated Profile Menu Program Cross-Reference
esaps_det	E-Signature Activated Profile Structure
esap_mstr	E-Signature Activated Profile
escat_mstr	E-Signature Category Master
escd_det	E-Signature Category Detail
escf_ref	E-Signature Category Filter Cross-Reference
escx_det	E-Signature Category XML Detail
esfil_mstr	E-Signature Filter
esgc_ref	E-Signature Group Category Cross-Reference
esig_mstr	E-Signature Master
esrec_det	E-Signature Record List
eswpfc_det	E-Signature Workbench Profile Filter Criteria
eswpfil_det	E-Signature Workbench Profile Filter
eswpfs_det	E-Signature Workbench Profile Filter Set
eswpf_det	E-Signature Workbench Profile Field
eswpmp_ref	E-Signature Workbench Profile Menu Program Cross-Reference
eswps_det	E-Signature Workbench Profile Structure
eswp_mstr	E-Signature Workbench Profile
exru_usage	Exchange Rate Usage
fal_mstr	Fixed Asset Location Code Master
fas_ctrl	Fixed Asset System Control
fldf_mstr	Field Default Master
flh_mstr	Field Help Program Master
hlp_mstr	Help Master
lblc_ctrl	Label Control
lng_mstr	Language Master
lngd_det	Language Detail
max_mstr	Maximums Master
maxt_det	Maximum Table Detail
mfrel_mstr	Master Table Relationships
mnd_det	Menu Detail
mnt_det	Menu Title Detail
msg_mstr	Message Master
pr_mstr	Printer Master
prd_det	Printer Detail
qaddb_ctrl	Database Control for QADDB
sbd_det	Sub-Account/Account Validation Detail
syp_mstr	Sync Profile Master
sypd_det	Sync Profile Detail
sypj_det	Sync Profile Join Detail

Table 15.1 — *Non-Domained Tables* — (Page 2 of 3)

Table	Description
syps_det	Sync Profile Subscription Detail
sytf_mstr	Sync Table-Field Master
tax_mstr	Tax Master
taxd_det	Tax Detail
typ_mstr	Fixed Asset Type Master
tzo_mstr	Service/Support Time Zone Master
tzod_det	Service/Support Time Zone Detail
ufd_det	User Function Key Detail
upd_det	Printer/User Detail
url_mstr	URL Master Table
usg_det	Application Usage Detail
uslh_hist	User Logon History
uspw_hist	User Password History
usr_mstr	User Master
usrc_ctrl	User Control
usrg_mstr	User Group Master
usrl_det	User Licensed Application Detail
vt_mstr	Value Added Tax Master
vtc_ctrl	Value Added Tax Control

Table 15.1 — *Non-Domained Tables* — (Page 3 of 3)

Programs that Update Cross-Domain Data

Some functions update data that is shared across domains. Table 15.1 lists functions that update this kind of data. The corresponding reports and browses also display shared data, but are not included in this table. For example, if Country Code Maintenance updates shared data, you can assume that Country Code Browse and Report display shared data.

Table 15.2
Programs Updating System-Wide Data

Menu No.	Program Description	Name
2.14.1	Country Code Maintenance	adctrymt.p
3.21.19	Transaction Numbering Report	ictrnrp.p
17.22	Operations Numbering Report	reopnrp.p
18.16	Operations Numbering Report	reopnrp.p
26.6	Exchange Rate Relationship Maintenance	mcdexrmt.p
26.7	Derived Exchange Rate Calculation	mcdexrcc.p
35.13.13	Transmission Group Maintenance	edtgmt.p
35.13.19	HTTP Adapter Maintenance	edhttpmt.p
35.15.6	Exchange Definition Maintenance	edxfmt.p
35.15.10	Application Definition Maintenance	edmfmt.p

Table 15.2 — *Programs Updating System-Wide Data* — (Page 1 of 4)

Menu No.	Program Description	Name
35.15.13	Implementation Definition Maint	edmimt.p
35.15.17	Transformation Definition Maint	edtrmt.p
35.15.21	ECommerce Function Maintenance	edtrfmt.p
36.2.1	Drill Down/Lookup Maintenance	mgdlfhmt.p
36.2.4	User Tool Maintenance	mgtoolmt.p
36.2.6	Menu Substitution Maintenance	mgmsmt.p
36.2.13	Browse Maintenance	mgbwmt.p
36.2.18	View Maintenance	mgvwmt.p
36.3.1	User Maintenance	mgurmt.p
36.3.3	User Password Maintenance	mgurmtp.p
36.3.4	User Group Maintenance	mgurgpmt.p
36.3.10	Menu Security Maintenance	mgpwmt.p
36.3.11	Menu Security Change	mgpwcg.p
36.3.21.1	Program Information Maintenance	mgpgmimt.p
36.3.21.3	Rule Maintenance	mgrulemt.p
36.3.21.5	Constant Maintenance	mgcnstmt.p
36.3.21.7	Profile Maintenance	mgprofmt.p
36.3.21.9	Profile Program Maintenance	mgprpgmt.p
36.3.21.11	Profile Program Rule Maintenance	mgpprlmt.p
36.3.21.14	Group Maintenance	mggrpmt.p
36.3.21.16	Group Constants Values Maint	mggrcnmt.p
36.3.21.18	User Group Security Maintenance	mgusrsmt.p
36.3.21.20	User Constants Value Maintenance	mguscnmt.p
36.3.21.24	Desktop Security Control	mgsecpm.p
36.3.21.23.21	Browse UI Records Maintenance	mgusrbmt.p
36.3.24	Security Control	mgurpmmt.p
36.4.1	Language Code Maintenance	mglmmt.p
36.4.3	Language Detail Maintenance	mglngumt.p
36.4.4	Menu System Maintenance	mgmemt.p
36.4.7	Message Maintenance	mgmsgmt.p
36.4.11	User Function Maintenance	mgufmt.p
36.4.13	Field Help Maintenance	mgflhusr.p
36.4.17.1	Label Master Maintenance	gplblmt.p
36.4.17.5	Label Detail Maintenance	gplbldmt.p
36.4.17.24	Label Control	gplblpm.p
36.4.18	Field Help Dump	mgfldmp.p
36.4.19	Field Help Load	mgflld.p
36.4.20	E-Mail Definition Maintenance	mgemmt.p
36.6.1	Database Connection Maintenance	mgdcmt.p
36.6.13	Database Connect	mgdccn.p

Table 15.2 — *Programs Updating System-Wide Data* — (Page 2 of 4)

Menu No.	Program Description	Name
36.6.15	Database Disconnect	mgdcdc.p
36.8.16	Export/Import Document Query	qqbr.p
36.8.17	Export/Import Document Report	qqierp.p
36.8.18	Dump Export/Import Docs to File	qqwrt.p
36.8.22.1	Synchronization Profile Maintenance	qqsypmt.p
36.8.22.3	Sync Table–Field Maintenance	qqsytfmt.p
36.8.22.8	Synchronization Mass Export	qqsymsex.p
36.10.1	Domain Maintenance	mgdommt.p
36.10.13	Change Current Domain	mgdomchg.p
36.12.13.1	Audit Trail Group Maintenance	attgmt.p
36.12.13.4	Audit Workbench Refresh	atwpref.p
36.12.13.5	Audit Workbench Profile Maintenance	atwpmt.p
36.12.13.8	Audit Profile Activation	atwpact.p
36.12.13.11	Audit DB Maintenance	atdbmt.p
36.12.13.23	Audit Trail Creation Process	atttpui.p
36.12.13.24	User Accountability Control	atpm.p
36.12.14.1	E-Signature Group Maintenance	escgmt.p
36.12.14.4	E-Signature Workbench Refresh	eswpref.p
36.12.14.5	E-Sig Workbench Profile Maint	eswpmt.p
36.12.14.8	E-Signature Profile Activation	eswpact.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
36.13.1	Printer Type Maintenance	mgmgmt04.p
36.13.2	Printer Setup Maintenance	mgmgmt05.p
36.13.4	Printer Default Maintenance	mgupmt.p
36.13.13	Print Queue Maintenance	mgmgmt07.p
36.14.3	Batch Request Detail Maintenance	mgbcdmt.p
36.14.13	Batch Request Processor	mgbatch.p
36.16.10.1	License Registration	lvreg.p
36.16.13	Sequence Maintenance	mgsqmt01.p
36.16.17	Database Sequence Initialization	utsequp.p
36.16.22.1	Multiple Time Zones Maintenance	fstzomt.p
36.16.22.13	Multiple Time Zone Load Utility	uttzld.p
36.19.1	AppServer Service Maintenance	mgasmt.p
36.20.10.1	User Option Maintenance	mgusromt.p
36.20.10.3	User Option Telnet Maintenance	mgusrtmt.p
36.20.10.8	Menu URL Maintenance	mgurlmt.p
36.20.10.11	Browse URL Maintenance	mgburlmt.p
36.20.10.15	Session Master Maintenance	mgsessmt.p

Table 15.2 — *Programs Updating System-Wide Data* — (Page 3 of 4)

Menu No.	Program Description	Name
36.22.1	Exit to Operating System	mgoscall.p
36.22.3	Program Execute	mgmgmt24.p
36.22.4	Program/Text File Display	mgfdsply.p
36.22.13	Disk Space Inquiry	mgdfds.p
36.24	Database Control	mgdbpm.p

Table 15.2 — *Programs Updating System-Wide Data* — (Page 4 of 4)

Default System Domain Data

Table 15.3 lists database tables containing data that is copied when a new domain is created.

Table 15.3
Tables Copied for New Domain

Table	Description
acdf_mstr	Account Default Master
ad_mstr	Address Master
apc_ctrl	Accounts Payable Control
arc_ctrl	Accounts Receivable Control
bic_ctrl	Service/Support Contract Billing Control
bk_mstr	Bank Master
bl_ctrl	Master Bill of Lading Control
cac_ctrl	Service/Support Call Master Control
caq_mstr	Service/Support Call Queue Master
cas_mstr	Service/Support Call Status Master
cc_mstr	Cost Center Master
cd_det	Master Comments
cfc_ctrl	Cash Flow Control
clc_ctrl	Compliance Control
cmc_ctrl	Customer Control
co_ctrl	General Ledger (Company) Control
code_mstr	Generalized Code Master
cr_mstr	Code Range Master
cs_mstr	Cost Set Master
drp_ctrl	Distribution Requirements Planning Control
egc_ctrl	Service/Support Engineer Schedule Control
emc_ctrl	Employee Control
es_mstr	Service/Support Escalation and Repair Master
esc_ctrl	Service/Support Escalation Control
esh_mstr	Service/Support Engineer Schedule Master
ess_mstr	Service/Support Engineer Status Master
et_ctrl	EMU Control

Table 15.3 — *Tables Copied for New Domain* — (Page 1 of 2)

Table	Description
fac_ctrl	Final Assembly Control
famt_mstr	Fixed Asset Method Master
gl_ctrl	Domain/Account Control
icc_ctrl	Inventory Control
iec_ctrl	Import/Export Control
ls_mstr	Address List Detail
mfc_ctrl	Control Work Table
mrpc_ctrl	Material Requirements Planning Control
opc_ctrl	Shop Floor Operation History Control
pcc_ctrl	Product Change Control
pgc_ctrl	Service/Support Paging Control
pic_ctrl	Pricing Control
pj_mstr	Project Master
pl_mstr	Product Line Master
poc_ctrl	Purchase Order Control
qcc_ctrl	Quality Order Control
qoc_ctrl	Sales Quotation Control
rmc_ctrl	Return Material Authorization Control
rnd_mstr	Rounding Method Master
rpc_ctrl	Repetitive Control
rsn_ref	Reason Code Master
sac_ctrl	Service Contract Control
sb_mstr	Sub-Account Master
sbc_mstr	Service/Support Billing Cycle Master
sc_mstr	Cost Simulation Master
shop_cal	Shop Calendar
soc_ctrl	Sales Order Control
spc_ctrl	Salesperson Control
src_ctrl	Service Request Control
sroc_ctrl	Service/Repair Order Control
sv_mstr	Service Agreement Terms and Conditions Master
svc_ctrl	Service/Support Management Control
trl_mstr	Trailer Master
tx2_mstr	Tax Master
txc_ctrl	Tax Control
vdc_ctrl	Supplier Control
woc_ctrl	Work Order Control

Table 15.3 — *Tables Copied for New Domain* — (Page 2 of 2)

Chapter 16

Using Q/LinQ with Multiple Domains

This chapter describes modifications and enhancements to Q/LinQ to support synchronization among multiple domains and cross-domain administrative activities.

Synchronizing Data

You can use features of Q/LinQ to synchronize static data such as item master data among multiple domains, both within a single database and between multiple, distributed databases. The data fields—called the *payload*—and the specific records to be updated—the *filter* or selection criteria—are specified in a *synchronization profile*.

When data included in a synchronization profile changes—either through addition, deletion, or modification—the event is captured using database *schema trigger*s, which publish the captured and filtered data to Q/LinQ as export synchronization documents.

Each document represents a single add, change, or delete action for a specific master table record. These documents can be viewed and reported on in the same way as other Q/LinQ documents.

See *External Interface Guide: Q/LinQ* for details on how Q/LinQ works.

Data Flow

The flow of data during synchronization varies depending on the method used.

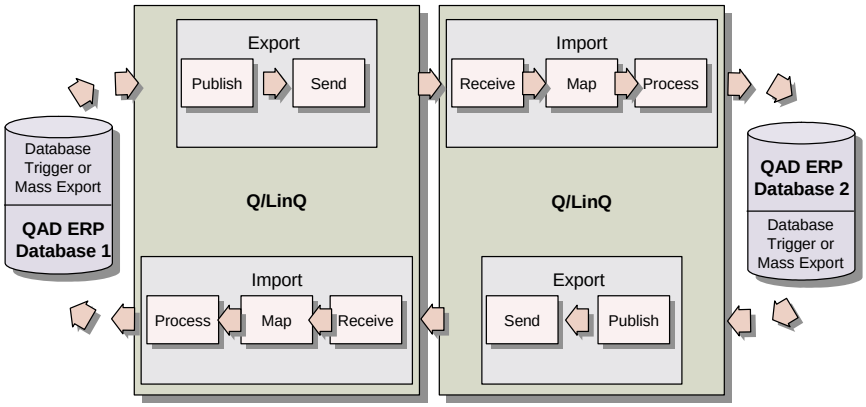
- Using one of the Q/LinQ data communications APIs is required when domains are distributed across multiple databases. An API can also be used when a single database has many domains.
- A streamlined forwarding method can be used for synchronizing data among domains in a single database. This is known as *intra-database forwarding*.

Data flow is simpler using intra-database forwarding.

Synchronization Through Q/LinQ Stream or Messaging API

Figure 16.1 illustrates the flow of data during synchronization between domains that communicate with each other through one of the Q/LinQ APIs.

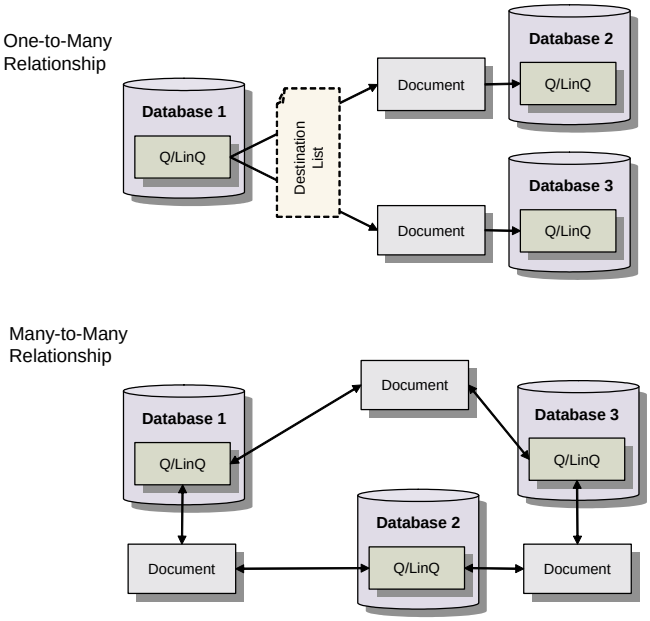
Fig. 16.1
Data Flow Between Databases Using APIs



When a table included in a synchronization profile is updated in a source domain, Q/LinQ publishes the data to an export document and exports that document to Q/LinQ in a destination database. The destination Q/LinQ receives the document, maps the data, and processes the data in the destination domain.

Figure 16.2 illustrates synchronization relationships among the domains.

Fig. 16.2
Relationships Among Domains in Different Databases



Source and destination domains can be in one-to-one, one-to-many, and many-to-many relationships. A source domain can synchronize different tables with different destination domains or different fields from the same table with different destination domains. In a one-to-many relationship, a destination list can manage the multiple export destinations.

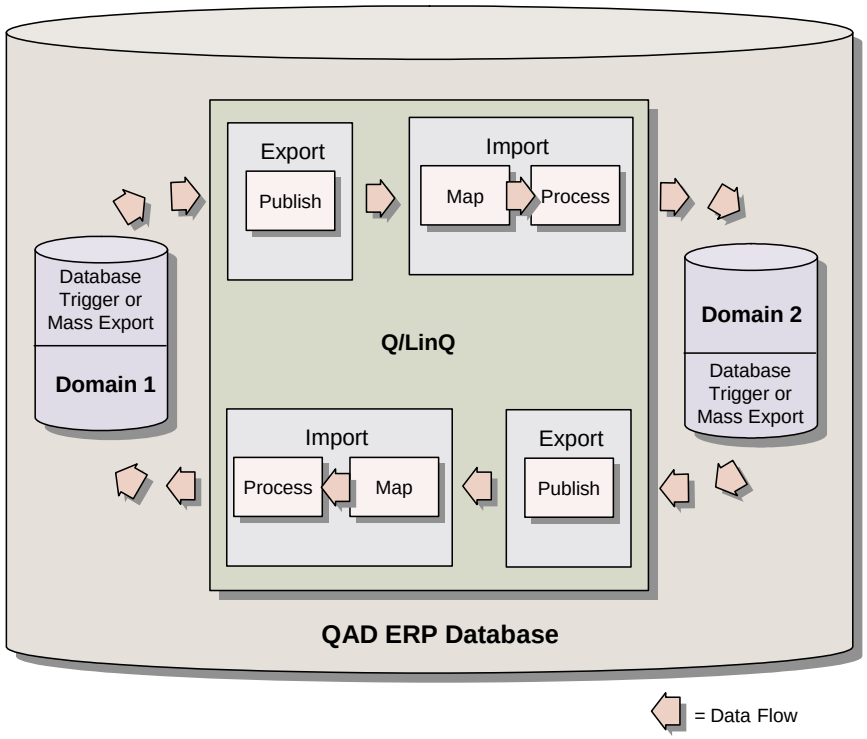
Note Although the figures illustrate domains in different databases, this method can also be used for domains within a single database.

Synchronization Through Q/LinQ Forwarding

Figure 16.3 illustrates the flow of data during synchronization between domains in a single database using Q/LinQ intra-database forwarding.

Note This method of synchronization can be used with domains in a single database only.

Fig. 16.3
Data Flow Among Domains Using Forwarding

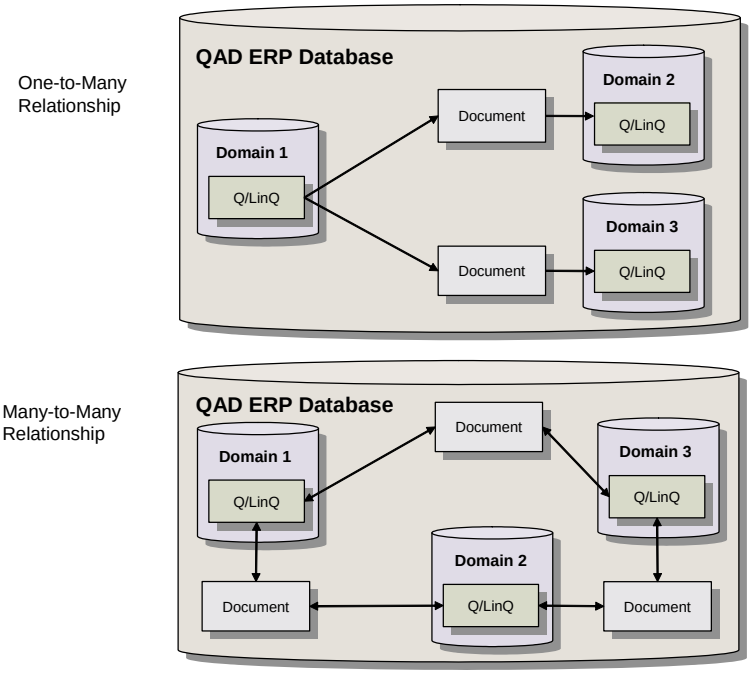


When a table included in a synchronization profile is updated in a source domain, Q/LinQ publishes the data to an export document and places it directly in the import queue of the destination domain. Q/LinQ then maps and processes the data in the destination domain.

Note In this scenario, one installed instance of Q/LinQ manages documents in all domains without establishing a communications link between them. In addition, the number of steps is reduced since the sending and receiving steps used between databases are not required.

Figure 16.4 illustrates the relationships among domains in a single database where Q/LinQ forwarding is used.

Fig. 16.4
Source and Destination Domain Relationships



Note Figure 16.4 illustrates relationships only; one instance of Q/LinQ in the database manages all these relationships.

Source and destination domains can be in one-to-one, one-to-many, and many-to-many relationships. A source domain can synchronize different tables with different destination domains or different fields from the same table with different destination domains.

Note When synchronizing among domains using Q/LinQ forwarding, destinations lists cannot be used. Instead you can use Export/Import Specification Copy (36.8.1.4) to streamline the setup of similar specifications. See “Copy Import and Export Specifications” on page 278.

Synchronization Documents

The records in export documents are published in Q/LinQ triplet format from data captured and filtered as specified by synchronization profiles. Each data field is represented with up to three character tokens:

```
[<context>]<name>=<value>
```

Where the token definitions are:

`<context>`	An optional qualifier used to identify the table name
`<name>`	The name of the application field
`<value>`	The value of the field expressed as an ASCII string

Each triplet is terminated by an end-of-line (carriage return or line feed) or a delimiter defined in the Data Mapping Parameters frame of Register External Applications (36.8.1.1), Export Specification Maintenance (36.8.1.2), or Import Specification Maintenance (36.8.1.3). The default delimiter is the pipe symbol (|).

Specifying Actions

Each document also includes a field indicating the type of action it represents:

`action=A` Add the data to the destination domain.

`action=C` Change the data in the destination domain.

`action=D` Delete the data from the destination domain.

For change actions, only net changes to a record are exported. These are the fields that have changed since the last maintenance transaction.

Example The published document includes this line when only the order quantity for the item has changed:

```
[pt_mstr]|action=C|pt_part=10-100-A|pt_ord_qty=120
```

Identifying Records Across Databases

Some tables that can be synchronized lack a meaningful unique key for identifying records between databases. For example, the price list detail table (pid_det) uses a database sequence (pid_list_id) in its unique key and database sequences are not meaningful across databases.

When a table lacks a meaningful unique key, the published document includes key information from a related master record. With this combination of master record data and detail record data, the destination database can identify the appropriate record to update.

Example This document for making price list detail changes includes unique identifying information from the price list master record:

```
[pi_mstr]|pi_list=test01|pi_list_id=100051|pi_cs_type=9|
pi_cs_code=qadall|pi_part_type=6|pi_part_code=qadall|
pi_curr=USD|pi_um=EA|pi_start=01/04/00

[pid_det]|action=C|pid_amt=50|pid_list_id=100051|
pid_qty=5
```

Note Q/LinQ processes this data the same way regardless of whether it is being published to another database or another domain within the current database. This is true even though the sequence number would be meaningful in the context of a single database.

Moving Data Between Domains

You can use any of the following Q/LinQ-supported methods when synchronizing data between domains within a single database or in connected databases:

- Intra-database forwarding (within a single database only)
- Q/LinQ-to-Q/LinQ Adapter (stream API)
- Messaging API
- File write/read

See “Set Up Communication Between Databases” on page 268 for details.

The recommended method for communicating synchronization data between domains is the Q/LinQ provided adapter, `qqqq2qq.p`, which is implemented using the Q/LinQ stream API. When all domains exist within a single database, using `qqqq2qq.p` streamlines setup, since you do not need to execute either the send or receive step; documents are published directly to the import queue of destination domains using Q/LinQ document forwarding capabilities.

When domains exist in separate databases, using this `qqqq2qq.p` adapter ensures a robust connection and also reduces the amount of setup required. With this adapter, sending and receiving documents is combined into one step. Without it, you must always execute a separate send and receive step to accommodate the message-oriented middleware that mediates the communication.

The Q/LinQ-to-Q/LinQ adapter can also be used to communicate between domains located in the same database, in place of intra-database forwarding. In this case, changes to data in the source domain are logged as Q/LinQ export documents, then re-imported into the destination domains using Q/LinQ send or receive functions. This approach adds run-time steps that are not required with intra-database forwarding, but has significant advantages related to performance and logging:

- Performance. With intra-database forwarding, Q/LinQ import documents are written into all the receiving domains when the user updates the data in the source domain. When there are many receiving domains, updates will take longer to complete.

  In one-to-many scenarios where a single source domain must synchronize with many receiving domains within its database, intra-database forwarding could adversely affect end-user performance in the source domain.

  With the Q/LinQ-to-Q/LinQ adapter, you can create a single destination list for the receiving domains. With lists, only a single export document is logged for all of receiving domains when the data is updated in the source domain.
- Logging. The Q/LinQ-to-Q/LinQ adapter provides a built-in audit trail of the data exported from each domain as well as the data imported into each domain. With intra-database forwarding, only import logs are written.

Note While it is possible to configure Q/LinQ to use file write/read to communicate between domains, this is not recommended.

For file write/read, users must provide their own batch control procedures for data integrity and loss protection. It is possible to read files continuously into Q/LinQ with a user-written polling procedure.

Table 16.1 summarizes the differences between the requirements for different communication approaches to data synchronization.

Table 16.1
Communication Approaches

Communication Approach	Register as External Application	Use Destination Lists	Send from Source	Receive into Destination	Logging
Intra-Database Forwarding	No	No	No	No	Destination domain only
Q/LinQ-to-Q/LinQ Adapter (`qqqq2qq.p`) of Stream API	Yes	Yes	Either send or receive	Either send or receive	Source and destination domains

Communication Approach	Register as External Application	Use Destination Lists	Send from Source	Receive into Destination	Logging
Messaging API (`qqmsgapi.p`)	Yes	Yes	Yes	Yes	Source and destination domains
File Write/Read	No	No	Yes	Yes	Source and destination domains

Data Mapping

In the source domain, Q/LinQ creates synchronization export documents in triplet format. In the destination domain, Q/LinQ uses table-specific mapping programs to map the data to CIM format.

See Table 16.3, “Synchronization Data Mapping Programs,” on page 276.

If site codes are not the same in different domains, Q/LinQ checks any values for pt_site defined in Code Mapping Maintenance (36.8.1.20) and maps them appropriately.

See “Create Optional Code Mappings” on page 269.

For the programs that update synchronized tables, Q/LinQ can process batch delete transactions through the CIM interface. The batch delete functionality is enabled only when the programs are accessed through a batch process; it is not enabled when the programs are used interactively. The one-character, unlabeled batch delete field is not visible to users, requires no changes to the user interface, and does not break existing CIM data files. Q/LinQ captures the delete action and the response to the Please Confirm Delete prompt. Use Ctrl+F in any of these programs to confirm that batch delete is enabled.

See Table 16.2 on page 253.

For more information on batch delete, see Chapter 6, “CIM Interface,” on page 65.

Tables to Synchronize

Table 16.2 lists the database tables that can be updated using data synchronization features of Q/LinQ. Also listed are the programs that are typically used to update the tables interactively.

The column labeled Required Table lists tables associated with the table being synchronized. These tables contain data that is validated during the update of the first table. If the data does not exist in the target domain, the synchronization will fail. You should ensure that data in prerequisite tables is also synchronized.

Table 16.2
Tables to Synchronize

Table	Table Description	Menu Label	Program	Req. Table
ac_mstr	Account Master	Account Code Maintenance	glacmt.p	code_mstr cu_mstr al_mstr fm_mstr
acdf_mstr	Account Default Master	Currency Account Maintenance	mccuacmt.p	cu_mstr ac_mstr al_mstr sb_mstr cc_mstr
an_mstr	Analysis Code Master	Analysis Code Maintenance	ppacmt.p	anl_det ans_det
anl_det	Analysis Code Link Detail	Analysis Code Link Maint	ppacln.p	an_mstr ans_det
ans_det	Analysis Code Selection Detail	Analysis Code Select Maint	ppacrl.p	an_mstr anl_det
bom_mstr	Product Structure Bill of Material	Product Structure Code Maint	bmmamt.p	
cc_mstr cr_mstr	Cost Center Master Code Range Master	Cost Center Code Maintenance	glccmt.p	ac_mstr al_mstr sb_mstr
cd_det	Master Comments	Master Comment Maintenance	gpcmmt.p	lng_mstr
cm_mstr ad_mstr ls_mstr	Customer Master Address Master Address List Master	Customer Maintenance	adcsmt.p	code_mstr ctry_mstr sp_mstr ac_mstr al_mstr sb_mstr cc_mstr cu_mstr si_mstr lng_mstr pi_mstr txz_mstr ct_mstr fr_mstr ft_mstr csbd_det
cm_mstr ad_mstr ls_mstr	Customer Master Address Master Address List Master	Customer Ship-To Maintenance	adstmt.p	code_mstr ctry_mstr lng_mstr txz_mstr
code_mstr	Generalized Codes Master	Generalized Codes Maintenance	mgcodemt.p	
cp_mstr	Customer Item Master	Customer Item Maintenance	ppcpmt.p	cm_mstr ad_mstr pt_mstr
cs_mstr	Cost Set Master	Cost Set Maintenance	csmsmt.p	

Table 16.2 — *Tables to Synchronize* — (Page 1 of 4)

Table	Table Description	Menu Label	Program	Req. Table
cu_mstr	Currency Master	Currency Maintenance	mccumt.p	
dpt_mstr	Department Master	Department Maintenance	rwdpmt.p	ac_mstr al_mstr sb_mstr cc_mstr
en_mstr	Entity Master	Entity Code Maintenance	glenmt.p	cu_mstr
exr_rate	Exchange Rate	Exchange Rate Maintenance	mcexrmt.p	cu_mstr
fcs_sum	Forecast Summary	Forecast Maintenance	fcfsmt01.p	pt_mstr si_mstr
glc_cal	General Ledger Calendar	GL Calendar Maintenance	glcalmt.p	glcalmt.p
is_mstr isd_det	Inventory Status Master Inventory Status Detail	Inventory Status Code Maint	icstmt.p	code_mstr
ls_mstr	Address List Master	Address List Type Maintenance	adlsmt.p	ad_mstr
pc_mstr	Price List Master – Purchasing	Price List Maintenance	pppcmt.p	cu_mstr pl_mstr pt_mstr um_mstr
pi_mstr pid_det	Price List Master Price List Detail – Sales	Price List Maintenance	pppimt.p	an_mstr cu_mstr um_mstr ac_mstr al_mstr sb_mstr cc_mstr pj_mstr
pl_mstr	Product Line Master	Product Line Maintenance	ppplmt.p	code_mstr ac_mstr al_mstr sb_mstr cc_mstr
ps_mstr	Product Structure Master	Product Structure Maintenance	bmpsmt.p	pt_mstr code_mstr

Table 16.2 — *Tables to Synchronize* — (Page 2 of 4)

Table	Table Description	Menu Label	Program	Req. Table
pt_mstr	Item Master	Item Master Maintenance	ppptmt.p	code_mstr um_mstr pl_mstr pcl_mstr loc_mstr alm_mstr is_mstr csim_mstr vd_mstr si_mstr ssm_mstr ssd_det ro_det bom_mstr cs_mstr
ro_det	Routing Operation Detail	Routing Operation Maintenance	rwromt.p	wc_mstr code_mstr vd_mstr pt_mstr
sb_mstr cr_mstr	Sub-Account Master Code Range Master	Sub-Account Code Maintenance	glsbmt.p	ac_mstr al_mstr
si_mstr	Site Master	Site Maintenance	icsimt.p	en_mstr is_mstr vd_mstr ac_mstr al_mstr sb_mstr cc_mstr
spt_det sct_det	Cost Simulation Item Detail Cost Simulation Total Detail	Item-Element Cost Batch Load	ppcsbtld.p	pt_mstr si_mstr cs_mstr sc_mstr
um_mstr	Alternate Unit of Measure Master	Unit of Measure Maintenance	pppummt.p	pt_mstr
vd_mstr ad_mstr ls_mstr	Supplier Master Address Master Address List Master	Supplier Maintenance	advnmt.p	code_mstr ctry_mstr ac_mstr al_mstr sb_mstr cc_mstr bk_mstr cu_mstr lng_mstr pc_mstr ct_mstr txz_mstr csbd_det
vd_mstr ad_mstr ls_mstr	Supplier Master Address Master Address List Master	Supplier Remit-To Maintenance	adrtmt.p	code_mstr ctry_mstr txz_mstr

Table 16.2 — *Tables to Synchronize* — (Page 3 of 4)

Table	Table Description	Menu Label	Program	Req. Table
vp_mstr	Supplier Item Master	Supplier Item Maintenance	ppvpmt.p	pt_mstr vd_mstr ad_mstr um_mstr cu_mstr pc_mstr
wc_mstr	Work Center Master	Work Center Maintenance	rwwcmt.p	dpt_mstr

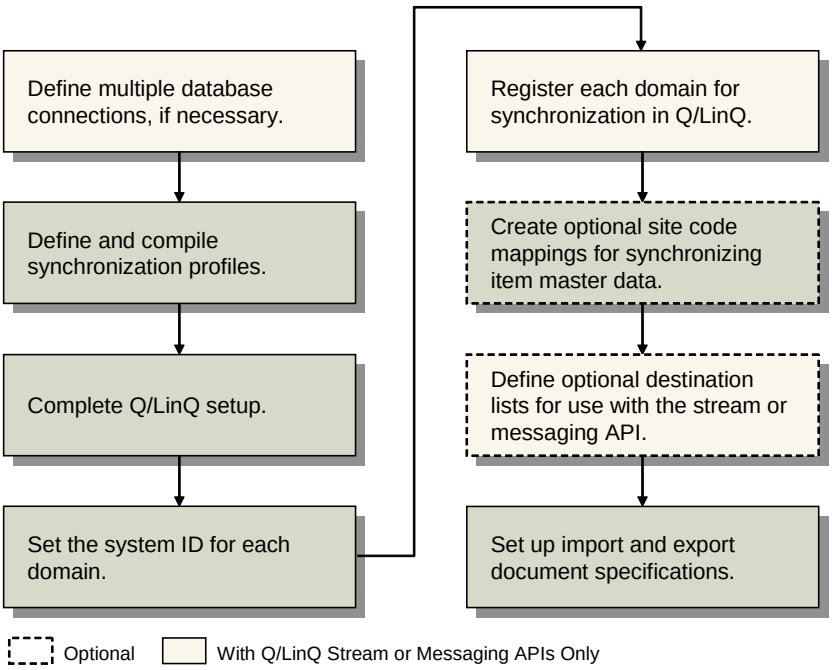
Table 16.2 — *Tables to Synchronize* — (Page 4 of 4)

Setting Up Synchronization

A number of setup steps are required to enable synchronization among multiple application domains. The steps vary depending on whether you are setting up synchronization between domains using Q/LinQ intra-database forwarding or using the Q/LinQ stream or messaging API. Setting up synchronization through intra-database forwarding requires fewer steps.

Figure 16.5 illustrates a typical work flow for setting up synchronization. See Figure 16.24 on page 279 for the processing work flow.

Fig. 16.5
Synchronization Setup Flow



In Figure 16.5, the steps with lighter shading are not required when setting up synchronization for domains within a single database using intra-database forwarding.

1. Define database connections for all databases participating in the synchronization using Database Connection Maintenance (36.6.1). This information lets Q/LinQ connect to domains in the other databases when required.
2. Define and compile synchronization profiles. These can be shared by all domains in a database. See page 258.
3. Complete the Q/LinQ setup. Complete information about setting up Q/LinQ can be found in E*xternal Interface Guide: Q/LinQ* and *Technical Reference: Q/LinQ*.
4. Set up system IDs for each domain in each database involved in synchronization. The system ID should normally be the same as the domain code. See page 265.
5. To synchronize between databases, register each external application domain in each domain that is involved in synchronization. This step is not needed if you are synchronizing data among domains in a single database using intra-database forwarding. However, if you use Q/LinQ APIs to synchronize data among domains in a single database, you must complete this step. See page 265.
6. Optionally define code mappings for sites in different domains. See page 269.
7. Optionally define destination lists in the source domain when the Q/LinQ stream or messaging API is used to link the domains. See page 270.
8. Create export specifications in source domains for outbound synchronization documents. See page 273.
9. Create import specifications in destination domains for inbound synchronization documents. See page 275.

Note You can streamline steps 8 and 9 by creating template specifications and copying them. See “Copy Import and Export Specifications” on page 278.

Review Tables and Fields for Synchronization

Sync Table–Field Maintenance (36.8.22.3) is populated with the names of the tables and fields that can be synchronized during database initialization. In most cases, the valid fields are those that can be updated directly by users through standard menu procedures.

If you have modified maintenance programs to display additional fields for update (for example, user-reserved table fields), you can use this procedure to make the new fields eligible for inclusion in synchronization profiles.

Important You should use security functions to limit access to this program. Incorrect entries can result in runtime Progress errors.

When you define a profile, Synchronization Profile Maintenance (36.8.22.1) validates tables and fields proposed for the profile against those in Sync Table–Field Maintenance.

To prepare for defining profiles, use Sync Table–Field Browse (36.8.22.4) to review the fields that can be synchronized from each table that can be synchronized.

Define Synchronization Profiles

Synchronization profiles specify which data—records and fields—to synchronize between domains and which types of data change—add, change, delete—to synchronize.

Synchronization profiles are not domain specific. This lets you share profiles among domains in a database. For example, if you need to keep the item master synchronized among three domains, you can use the same profile for each. If you do not want to share profiles, establish a naming convention to distinguish them.

Since profiles are defined without reference to a destination, they can also be used to implement synchronization with multiple destination databases.

This section covers general profile setup as well as setup for tables with dependencies.

Use Synchronization Profile Maintenance (36.8.22.1) to define profiles. Use Synchronization Profile Inquiry (36.8.22.2) to review the contents of synchronization profiles.

Default Profiles

To simplify profile setup, a set of default profiles is loaded during system initialization. A profile exists for each table that can be synchronized. These profiles support the synchronization of all records and eligible fields.

You can use these profiles they way they are; however, if you want to filter data based on some criteria, you can modify them to meet your requirements or create your own.

Note Even if you use the profiles delivered by QAD, they must be compiled. See “Compiling Profiles” on page 260.

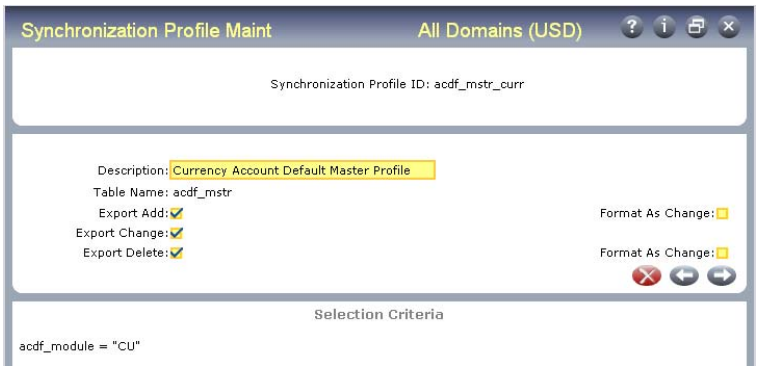
These profiles have the same name as their associated table. For example, the profile for synchronizing item master data is pt_mstr.

General Profile Setup

Synchronization profiles contain the following:

- The name of the table with data to be synchronized. Each profile can specify only one table. See Table 16.2 on page 253.
- Settings indicating which events (add, change, delete) create a synchronization document.
- Selection criteria specifying which records from the table should be synchronized. This is the filtering criteria.
- The fields in the specified table whose value will be included in the synchronization document. This is the payload.

Fig. 16.6
Synchronization Profile Maintenance (36.8.22.1)



Sync Profile ID. Enter a unique identifier for the synchronization profile.

Description. Enter up to 40 characters describing the synchronization profile.

Table Name. Enter the name of the table with fields to be included in the synchronization document. This table must be listed in Sync Table–Field Maintenance.

Note You cannot change this value after it has been defined.

Export Add. Enter Yes to export data records that now match the selection criteria of the synchronization profile where previously they did not match the criteria. A record may be new or it may now match the selection criteria because its value has been updated.

Format as Change. Enter Yes to format records selected by Export Add as changes when exporting. This field should be Yes only when the destination domains have already set up all possible records in advance. Set this field to Yes to avoid sending an add-record instruction to a domain where the record already exists.

Export Change. Enter Yes to export data records that are updated. This applies to records that previously met and still meet the selection criteria of the synchronization profile.

Export Delete. Enter Yes to export a delete instruction for a record with a value that no longer meets the selection criteria of the synchronization profile. The record may continue to exist in the source domain but its data is no longer exported for synchronization.

Format as Change. Enter Yes to format records selected by Export Delete as changes when exporting. Set this field to Yes when records cannot be deleted from the destination domain but record status must change, such as an item changing from an active to an obsolete state.

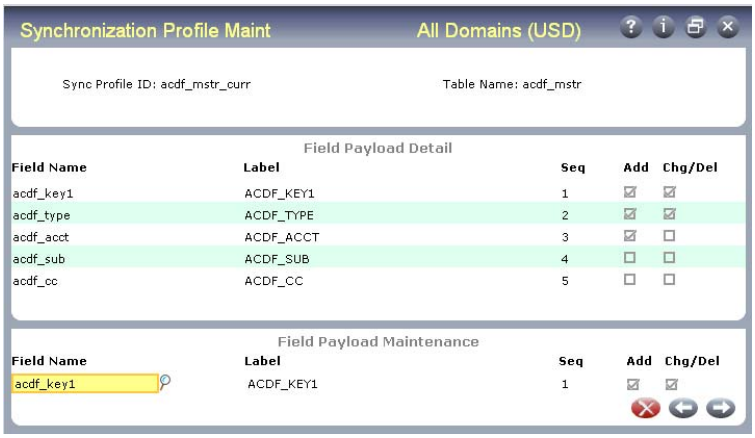
Selection Criteria. Enter the legal Progress 4GL syntax for selecting records to synchronize. Use this field to describe a subset of data records from the synchronization profile's database table, such as only items with an active status or only items in product line 100. The criteria is validated, interpreted, and processed as a Progress 4GL WHERE clause.

Use Progress functions or operators in any combination containing one or more levels of parenthesized expressions if necessary. Use only logical expressions; do not use IF, ELSE, WHERE, or other reserved words.

References can be made only to the table associated with the current synchronization profile, except inside CAN-FIND function invocations, which can reference other tables. The system compiles the entered expression and returns any Progress error messages as warning conditions to the user.

Blank causes all records in the table to be subject to synchronization.

Fig. 16.7
Synchronization Profile Maintenance, Field Payload Frames



The Field Payload Detail frame displays current settings for each field in the table.

Use the Field Payload Maintenance frame to add fields to the profile payload and modify or delete existing fields in the payload.

Field Name. Select the field to be added to, modified, or deleted from the profile payload.

Sequence. Enter the sequence of the field in the profile payload. This field does not affect processing. Use it to order the fields for convenience.

If a sequence number is not assigned or if a number is assigned that is greater than the number of fields in the profile, Q/LinQ assigns the next available number to the field. As you assign fields different numbers, Q/LinQ closes gaps in the sequence and renumbers all fields so that the highest sequence number is always the number of fields in the profile payload.

Required on Add. Enter Yes to always export the field for add actions, and to delay the publishing of the record until the field has been populated with a non-empty value. Use this to export key or other fields required to create a new record in the destination domains.

Example Always export product line (pt_prod_line) when adding item master (pt_mstr) records, and do not publish the new pt_mstr record until its product line has been assigned. Although it is not a pt_mstr key, a new record requires a value for the pt_prod_line field.

Require on Change or Delete. Enter Yes to always export the field for change or delete actions. Use this to export key fields used for locating records in the destination domain.

Example Always export item number (pt_part) when modifying item master (pt_mstr) records because it is the unique key for the pt_mstr table.

Compiling Profiles

After you add the fields you want included in the payload of the synchronization profile, leave the Field Payload Detail frame by pressing End. This causes Progress r-code to be automatically generated and compiled from the synchronization profile and stored in the Q/LinQ archive directory, as established in the Q/LinQ initialization file `qqapi.ini`.

If any compile errors occur—typically due to invalid selection criteria— Progress errors display as warning messages. The profile cannot be used until the r-code is compiled successfully from Synchronization Profile Maintenance.

Important Predefined synchronization profiles provided by QAD do not work until you press End from the Field Payload Detail frame of Synchronization Profile Maintenance for each of the profiles. This must be done to properly compile the r-code even when you do not modify the content of the profiles.

Setting Up General Header-Detail Tables

Related tables such as those with header-detail or parent-child relationships must be updated in the destination domain at the same time. For example, synchronizing analysis code link detail (anl_det) without synchronizing analysis codes (an_mstr) would not be meaningful. And synchronizing only analysis code link detail would not provide sufficient information for the destination domain to use the data.

Use this setup for synchronizing related tables:

- Use Synchronization Profile Maintenance to set up synchronization profiles with matching selection criteria for the header (parent) and detail (child) tables. For example:

Table	Selection Criteria in the Synchronization Profile
an_mstr	an_code>= “1000”, an_code<= “5000”
anl_det	anl_code>= “1000”, anl_code<= “5000”

- In the source domain, use Export Specification Maintenance to create export specification records that associate the header and detail synchronization profiles with the destination domain.
- In the destination domain, use Import Specification Maintenance to set up import specification records with document types that match those in the header and detail export specification records.

Setting Up Addresses

You can synchronize data for the following types of addresses:

- Customer
- Customer ship-to
- Supplier
- Supplier remit-to

Because of the way address data is stored in the database, updating one of these records can affect up to three tables:

- The table for the specific address type (cm_mstr, vd_mstr)
- The table for general address information (ad_mstr)
- The table that stores system and user-defined address list types (ls_mstr)

  Note Cross-references between ad_mstr records and cm_mstr records are stored in ls_mstr.

For each destination domain, ad_mstr and ls_mstr records are synchronized only when they are associated with customer or supplier records that are set up for synchronization.

Note You can choose to synchronize address list data, but only for types that are not system generated. See “Address List Type Table Setup” on page 263.

To streamline synchronization, it is best to have one address profile for each type of synchronized address: supplier, customer, ship-to, and remit-to. This provides more flexibility in controlling the events that require synchronization for each address type and also the fields that need to be synchronized.

Use the guidelines in the following sections for setting up address synchronization profiles in source domains. In each case, after setting up profiles, complete these steps:

1. In the source domain, use Export Specification Maintenance to create export specification records that associate the profiles with the proper destination domain.
2. In the destination domain, use Import Specification Maintenance to set up import specification records with document types that match those in the export specification records.

Note You can also create template specifications and use Export/Import Specification Copy (36.8.1.4) to quickly create similar records in multiple domains. See “Copy Import and Export Specifications” on page 278.

Customer and Address Master Table Setup

Set up synchronization profiles for the cm_mstr and ad_mstr tables. The selection criteria need not be the same since Q/LinQ automatically checks for association between customer and address records.

For customers, it is recommended that the following selection criteria be used for the ad_mstr profile:

```
ad_ref = “” and ad_type = “customer”
```

For customer ship-to addresses, use the following selection criteria for the ad_mstr profile:

```
ad_ref <> “”
```

For customer records, you should also set the Required on Add field to Yes for the cm_site field in the cm_mstr payload. This prevents the creation of source documents with incomplete information that might fail to be loaded in the receiving domain.

Supplier and Address Master Table Setup

Set up synchronization profiles for the vd_mstr and ad_mstr tables. The selection criteria need not be the same since Q/LinQ automatically checks for association between supplier and address records.

For suppliers, it is recommended that the following selection criteria be used for the ad_mstr profile:

```
ad_ref = “” and ad_type = “supplier”
```

For supplier remit-to addresses, use the following selection criteria for the ad_mstr profile:

```
ad_ref = “” and ad_type = “remit-to”
```

For suppliers, you should also set the Required on Add field to Yes for the vd_curr field in the vd_mstr payload. This prevents the creation of source documents with incomplete information that might fail to be loaded in the receiving domain.

Address List Type Table Setup

Only list types that have been assigned by users can be synchronized directly. System-assigned address list types are not synchronized since these list type are created automatically when the associated address records are created.

Note System-assigned list types include slsprsn, company, customer, enduser, ship-to, supplier, remit-to, dock, c/s_bank, our_bank, po-ship, carrier, and engineer.

Use this setup for synchronizing address list type records:

1. Set up synchronization profiles for the cm_mstr, vd_mstr, ad_mstr (ship-to, remit-to), and ls_mstr tables.
2. In the source domain, use Export Specification Maintenance to create export specification records that associate the ls_mstr, cm_mstr, vd_mstr, and ad_mstr synchronization profiles with the destination domain.
3. In the destination domain, use Import Specification Maintenance to set up import specification records with the document types that match those in the ls_mstr, cm_mstr, vd_mstr, and ad_mstr export specification records

Setting Up Bill of Material Profiles

You can synchronize only bill-of-material (BOM) records (bom_mstr) that are created, modified, or deleted using Product Structure Code Maintenance (13.1). Other options for creating bom_mstr records in the following programs are not supported at this time:

- Service BOM Code Maintenance (11.19.1)
- Formula Code Maintenance (15.1)

Use this setup for synchronizing BOM records:

1. Set up synchronization profiles for the bom_mstr table using the following selection criteria:
   `bom_formula = no and bom_fsm_type = “” and bom_batch = 0.0`
2. In the source domain, use Export Specification Maintenance to create export specification records that associate the bom_mstr synchronization profiles with the destination domain.
3. In the destination domain, use Import Specification Maintenance to set up import specification records with the document types that match those in the bom_mstr export specification records.

Setting Up Product Structure Profiles

You can synchronize only product structure records (ps_mstr) that are created, modified, or deleted using Product Structure Maintenance (13.5). Other options for creating ps_mstr records in the following programs are not supported at this time:

- Configured Structure Maintenance (8.1)
- Service Structure Maintenance (11.19.5)
- Alternate Structure Maintenance (13.15)
- Co/By-Product Maintenance (13.22.1, 15.12.1)

- Formula Maintenance (15.5)
- Process/Formula Maintenance (15.18)

Use this setup for synchronizing product structure records:

1. Set up synchronization profiles for the ps_mstr table using the following selection criteria:
   `ps_ps_code <> “J” and ps_ps_code <> “A” and ps_qty_type = “”`
2. In the source domain, use Export Specification Maintenance to create export specification records that associate the ps_mstr synchronization profiles with the destination domain.
3. In the destination domain, use Import Specification Maintenance to set up import specification records with the document types that match those in the ps_mstr export specification records.

Setting Up Routing Detail Profiles

You can synchronize only routing records (ro_det) that are created, modified, or deleted using Routing Maintenance (14.13.1). Other options for creating ro_det records in the following programs are not supported at this time:

- Service Routing Maintenance (11.19.17)
- Routing Maintenance–Rate Based (14.13.2)
- Formula Maintenance (15.13)
- Process/Formula Maintenance (15.18)

Use this setup for synchronizing routing records:

1. Set up synchronization profiles for the ro_det table using the following selection criteria:
   `ro_fsm_type = “”`
2. In the source domain, use Export Specification Maintenance to create export specification records that associate the ro_det synchronization profiles with the destination domain.
3. In the destination domain, use Import Specification Maintenance to set up import specification records with the document types that match those in the ro_det export specification records.

Complete Q/LinQ Setup

Install and set up Q/LinQ for each database that will export data to or import data from other databases for data synchronization.

See *External Interface Guide: Q/LinQ*.

Note Only one instance of Q/LinQ is required if you are going to synchronize data between domains in a single database.

You should limit access to most Q/LinQ programs—including the data synchronization programs—to administrators. This limited access is especially important for Sync Table–Field Maintenance (36.8.22.3), which identifies the tables and fields that can be synchronized. This table is populated during installation and should not be modified.

See “Assign Access by Menu” on page 168 for details on menu security.

Important Do not use Sync Table–Field Maintenance to add other tables or fields since these are not supported by the software and lead to runtime Progress errors.

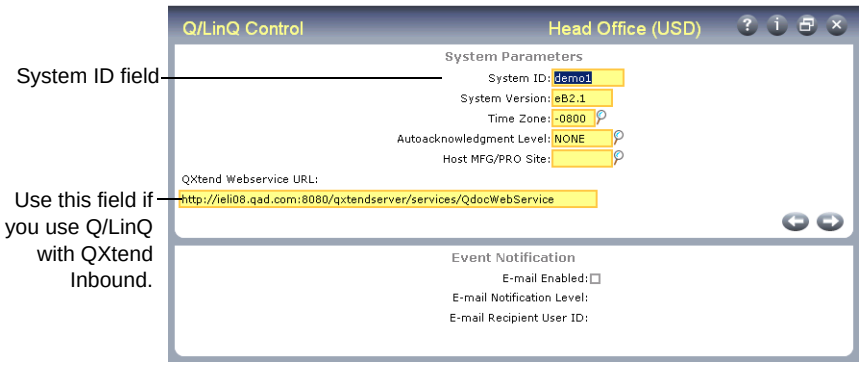
Set Up System IDs for Domains

The system ID defined in Q/LinQ Control (36.8.24) identifies the source application system on exported documents. This ID is typically set to the domain code. In a multiple-domain environment, this ensures that you can easily and consistently identify where documents originated.

When an import or export specification references a system ID in the current database that is not registered with Q/LinQ as an external application, Q/LinQ recognizes that synchronization is taking place with a domain in the current database, and copies the data across domains using intra-database forwarding.

When Q/LinQ Control is first accessed, system ID defaults to the current working domain.

Fig. 16.8
Q/LinQ Control (36.8.24), System ID Field



Q/LinQ validates that the system ID is unique within the database and is not the same as any other application ID defined in Register External Application (36.8.1.1).

If you have installed QXtend Inbound, specify the URL that identifies the QXtend server in your system (maximum 70 characters). The system uses this URL to locate the server when the import specification associated with a document indicates that it should be processed through QXtend.

See *Technical Reference: QXtend Inbound* for details.

Register Domains

Important This step is required only when you are synchronizing data with domains using the Q/LinQ-to-Q/LinQ adapter or Q/LinQ messaging API. It is not needed for intra-database forwarding.

In each source application domain, register each destination application domain as a Q/LinQ external application in Register External Application (36.8.1.1). Accept the Q/LinQ defaults except where noted. You must also create the corresponding registrations in each destination domain by registering each source domain.

See *External Interface Guide: Q/LinQ* for additional details.

The key fields for registering domains are Application ID and Access Code/Path:

- Each application ID identifies a domain in this or an external database. Set it to the value of system ID in Q/LinQ Control of the referenced domain.
- The Access Code/Path specifies the actual domain code as defined in Domain Maintenance (36.10.1). The system determines the database associated with the domain based on the value defined in Domain Maintenance. If the database is not the current one, database connection parameters are found in Database Connection Maintenance (36.6.1). See “Domain Maintenance” on page 11.

Note The Q/LinQ system ID is typically the same as the domain code; keeping them the same makes it simpler to identify document ownership.

Application ID Examples

The data you define in Register External Application is domain specific. If more than one domain shares data with another domain, you must complete the database registration tasks in each domain. This ensures that export and import documents are always sent to and received by the correct domain.

The following examples illustrate setting up IDs in different scenarios.

Example 1: Many-to-Many

Three possible combinations exist for three domains in three separate databases (`Domain1`, `Domain2`, `Domain3`) that each exchange data with all of the others. Use the domain system ID values as the application IDs when registering the databases/domains in Q/LinQ.

See Figure 16.2 on page 247.

In this Domain System ID	Register These Application IDs
domain1	domain2, domain3
domain2	domain1, domain3
domain3	domain1, domain2

Example 2: One-to-Many

A configuration of one central domain (`domainC`) and two peripheral domains in separate databases (`domainE` and `domainW`) has only two possible combinations. The sample application IDs and the domains where they must be registered are shown in this table.

See Figure 16.2 on page 247.

In this Domain System ID	Register These Application IDs
domainC	domainE, domainW
domainE	domainC
domainW	domainC

Register IDs

In Register External Application, enter the application ID for a domain in a synchronization relationship. Also enter a name and description for the synchronization represented by the application ID.

In the Select Data Update frame, enter Yes in these fields:

- Interface Control Parameters, to update e-mail settings
- Default Communications Parameters, to set up between-database communication
- Miscellaneous Defaults, to embed document control tags when using the stream or messaging API

Press Go to display the first frame selected for update. After making edits, press Go again to display the next frame selected for update. Press End at any time to return to the main program screen.

Fig. 16.9
Register External Application (36.8.1.1)

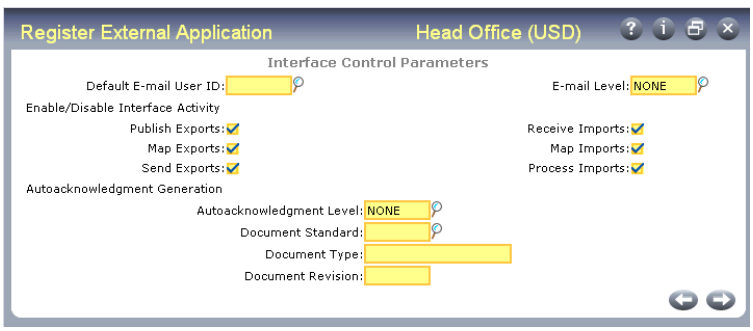
Set E-Mail Notification

The e-mail notification settings default from Q/LinQ Control (36.8.24). Set up application-specific e-mail notification in the Interface Control Parameters frame.

Enter the name and preferred e-mail notification level of the system administrator who is the default recipient of any Q/LinQ-generated e-mail messages about this synchronization relationship.

Press Go to display the next frame selected for update.

Fig. 16.10
Register External Application, Interface Control Parameters



Set Up Communication Between Databases

Synchronization documents can be passed between databases as text files or using one of the communication APIs. The recommended method is using `qqqq2qq.p`, discussed next.

See *External Interface Guide: Q/LinQ* for a discussion of exchanging documents as text files or with APIs.

Stream API

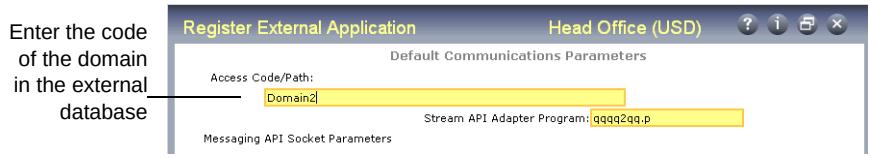
Use the synchronization adapter, `qqqq2qq.p`, to exchange data through a direct Q/LinQ-to-Q/LinQ Progress database connection. The Progress two-phase commit feature ensures the data integrity of the transmission.

Note The adapter can create a connection between two Progress databases or two Oracle databases (UNIX only); it cannot create a connection between a Progress database and an Oracle database.

In the Default Communication Parameters frame, enter the code of the domain in the external database in Access Code/Path. Q/LinQ determines the database associated with the domain by looking up the domain record defined in Domain Maintenance (36.10.1). If the database is not the current working database, then database switching is initiated.

Enter the `qqqq2qq.p` adapter name in Stream API Adapter Program.

Fig. 16.11
Default Communications Parameters: Stream API



Messaging API

The messaging API establishes a direct TCP/IP connection with Q/LinQ in an external database using a Q/LinQ-specific protocol that is independent of Progress client-server connections. This requires a port for each direction of the connection. This can be two different ports, or you can use the same for both import and export.

Note To use the messaging API, leave the Stream API Adapter Program field blank.

Use these guidelines for entering information in both the export and import fields of the Messaging API Socket Parameters frame:

- If the current database initiates the connection as the active caller, enter the host name or the Internet protocol (IP) address and the port number of the remote host, the listener.
- If the remote database initiates the connection and the current database is the passive listener, no host name, IP address, or port number is required.
- For Q/LinQ Initiates Connection:
  - Enter Yes to have the current database actively request a connection with the external database on the designated host. This means that the current database is the caller or client for the TCP/IP session.

- Enter No to have the current database monitor its local socket for a connection request from the external database. This means that the current database is the listener or server for the TCP/IP session.

Note Q/LinQ only initiates the connection; it does not start or stop any program in the external application.

As shown in Figure 16.12 and Figure 16.13, for two-way data synchronization, the import and export port designations must be complementary between domain pairs in connected databases.

Fig. 16.12
Communication Parameters: Domain 1, Database 1

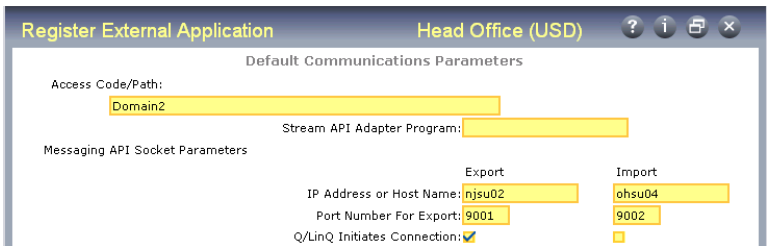
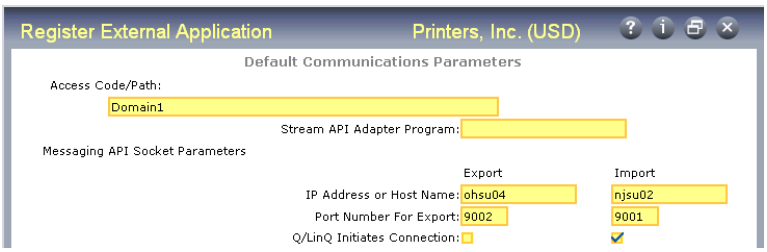


Fig. 16.13
Communication Parameters: Domain 2, Database 2



Miscellaneous Defaults

In the Miscellaneous Defaults frame, enter Yes in Precede Data with Tags to embed document control tags in front of the actual data when using the stream or messaging API to send documents to external databases. Control tags are always included when sending documents to a file.

See *External Interface Guide: Q/LinQ* for a discussion on document control tags.

Create Optional Code Mappings

For some variables or codes, values used in an external application may differ from those used in QAD Enterprise Applications. To accommodate these differences, you can use Code Mapping Maintenance (36.8.1.20) to define mappings used to translate data values when documents are exchanged.

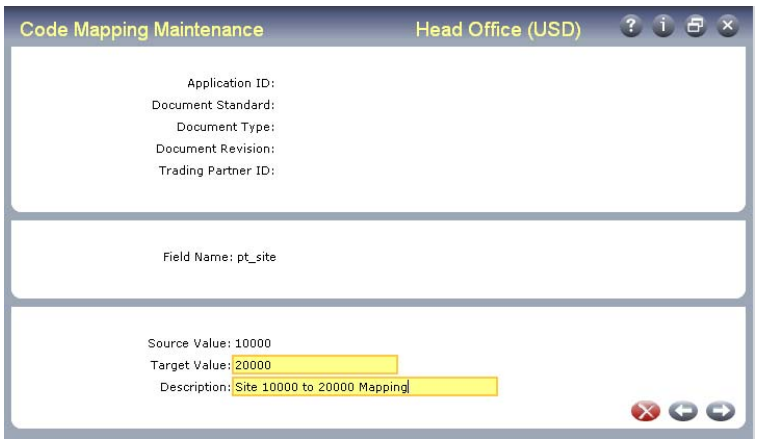
When synchronizing data, one particular case may require the use of code mappings. This is the site field (pt_site) associated with items in Item Master Maintenance (1.4.1). If this field is included in the synchronization profile, it must be mapped to a site value that is valid for the destination domain.

When you create a site, you can create a connection record in all active sites in the current database or connected databases. However, creating connection records is optional; it is possible that a site does not exist in all domains containing data being synchronized.

To prevent errors during synchronization of item master data, you can create a mapping for the site value in a source domain to a valid site value in the destination domain.

See “Associating Domains with Sites” on page 14.

Fig. 16.14
Code Mapping Maintenance (36.8.1.20)



To create a mapping for the Site field, follow these steps:

1. Leave the fields Application ID, Document Standard, Document Type, Document Revision, and Trading Partner ID blank. This ensures that the mapping applies to all synchronization documents.
2. Specify pt_site as the Field Name.
3. For Source Value, specify a site in the source domain associated with item data to be synchronized.
4. For Target Value, specify a valid site in the destination domain.
5. Enter an optional description of the mapping.

Define Destination Lists

Important This step applies only when you are synchronizing data between domains using the Q/LinQ-to-Q/LinQ adapter or Q/LinQ messaging API.

Destination lists are lists of application IDs defined in Register External Application that can be used for publishing, sending, and deleting/archiving documents. Destination lists are single level; they cannot be nested. In the context of data synchronization, they can be used to streamline the setup required for communicating with multiple external domains or applications.

Note the following restrictions on destination lists:

- They are used for exporting, but not for importing documents.
- They can be used for communicating between domains in the same or other databases, but not for synchronizing data among domains in a single database using Q/LinQ intra-database forwarding.

Using destination lists is optional, but they can facilitate data synchronization among domains in multiple databases or when there are many domains in a single database. Without them, each trigger or export event sends one document to each target domain. Destination lists let you send each event to a list of domains in this or external databases. This saves disk space and improves runtime performance because it creates only one copy of the export documents for management and storage rather than one for each destination.

Publishing and sending to destination lists are independent tasks:

- Documents published to an individual domain can be sent to that domain or to any destination list to which the domain belongs.
- Documents published to one list can be sent to individual domains or to another list where only the destinations common to both lists receive the document.

Note Documents can be published and sent to domains that are registered to use the stream or messaging APIs for communication. The messaging API can be used with destination lists for sending documents only on UNIX systems. The stream API can be used with destination lists for sending documents on both UNIX and Windows systems.

Use Destination List Maintenance (36.8.8) to define destination lists.

Fig. 16.15
Destination List Maintenance (36.8.8)

Destination List ID. Enter a unique name for the destination list.

The system verifies that this name is not currently used as the Q/LinQ system ID for any domain in the current database. System ID is defined for each domain in Q/LinQ Control. Also, this name cannot be the same as any registered application ID in the current domain.

Description. Enter up to 60 characters describing the destination list.

Application ID. Enter the ID of an application registered in Register External Application. For data synchronization, the application ID is the same as the system ID of the domain.

An error displays if the ID you enter is not registered with Q/LinQ as an external application in the current domain.

Effective Date In. Enter the date when the database's membership in the distribution list becomes effective.

Effective Date Out. Enter the date when the database's membership in the distribution list expires.

Set Up Document Specifications

Use Export Specification Maintenance (36.8.1.2) and Import Specification Maintenance (36.8.1.3) to identify particular documents and to set document-specific parameter values for exporting and importing. For exports, you can define interface control, document content, data mapping parameters, messaging, and miscellaneous parameters. For import, you define interface control, data mapping, and miscellaneous parameters.

Note When the specification is associated with a system ID that is not registered as an external application—indicating a domain within the current database—the frames for updating data mapping, messaging, and miscellaneous parameters for export specifications do not display. These values are not used for synchronizing data among local domains.

When specifications are associated with registered applications, many fields default from the values you specify in Register External Application. These defaults do not exist when creating specifications for use with domains in the same database.

For registered applications, you may be able to import and export documents without a specific specification in limited cases. In these cases, the values specified in Register External Application are used as is.

See “Register Domains” on page 265.

Matching Specifications to Documents

You can use up to five values to define an import or export specification: document standard, document type, document revision, application ID, and trading partner ID. The only required field is the document type. The system uses the following logic to find a specification to apply to a document:

1. It looks for one with an exact match for document standard, document type, document revision, application ID, and trading partner ID.
2. It looks for one with matching document standard, document type, document revision, application ID, and a blank trading partner ID.
3. It looks for one with matching document standard, document type, document revision, and blank application and trading partner IDs.

This lets you set up generic specifications that can apply to all documents of a certain type (and optional standard and revision) regardless of the particular application or trading partner associated with a document.

This can be useful when you set up specifications for synchronizing data among domains in separate databases. You can define a generic specification for a particular table (identified by the document type) and leave the application ID blank. Then copy this record to each affected domain using Export/Import Specification Copy. This lets you quickly set up similar export and import specification records.

See “Copy Import and Export Specifications” on page 278.

Required Specifications

Import specifications are required when:

- A document type must be routed to a specific, non-default program for processing.
- A document type must be mapped by a specific, non-default mapping procedure or mapping specification.

Export specifications are required when:

- A document type must be associated with a specific synchronization profile.
- A document type must be mapped by a specific, non-default mapping procedure or mapping specification.

Defining specifications for unique export documents supports flexibility in data synchronization. For example, you can:

- Export synchronization events for different tables to different destinations.
- Export synchronization events for different fields in specific tables to different destinations.

Register Export Specifications

Use Export Specification Maintenance to:

- Create an export specification for each synchronization document type to be exported to each domain.
- Associate a synchronization profile with each synchronization document type.
- Optionally associate a synchronization profile with a system ID, registered application, or destination list.

Accept the Q/LinQ defaults except where noted.

Fig. 16.16
Export Specification Maintenance (36.8.1.2)

Application or Destination List ID. Enter one of the following:

- The system ID of another domain in this database as defined in Q/LinQ Control (36.8.24)
- The ID of a domain in this or an external database as it is defined in Register External Application (36.8.1.1)
- The name of a destination list as defined in Destination List Maintenance (36.8.8)

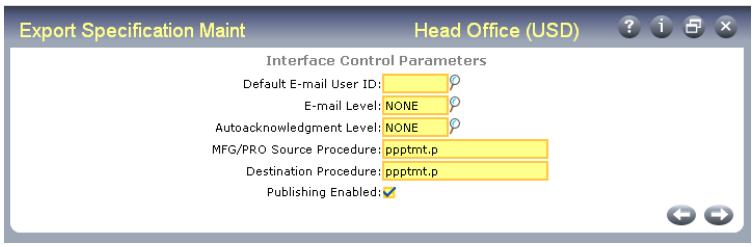
Leave blank if you want this specification to apply to all documents of a certain type, standard, and revision without regard to the associated application.

Document Standard. Enter a user-defined name such as sync_docs so that synchronization activity and documents can be easily segregated from other Q/LinQ documents and activities. This field is validated against generalized codes defined for field esp_doc_std.

Document Type. Enter a user-defined name for the type of data that is being synchronized. Use names that reflect the type of data, such as pt_part for item numbers, to facilitate browsing, reporting, and tracking specific data elements.

Document Revision and Trading Partner ID can be left blank. If they are used, Document Standard, Document Type, Document Revision, and Trading Partner ID must be a unique combination of values.

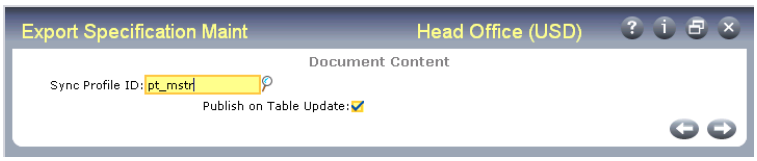
Fig. 16.17
Export Specification Maintenance, Interface Control Parameters



Default E-mail User ID, E-mail Level. Enter data for fields, as needed. For registered application IDs, these fields default from Register External Application.

Publishing Enabled. Enter Yes when the export specification is ready to be used by Q/LinQ. Enter No for Q/LinQ to ignore this specification.

Fig. 16.18
Export Specification Maintenance, Document Content



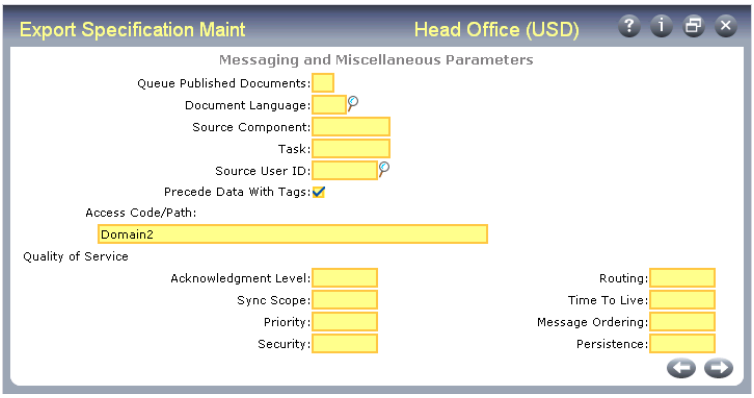
Profile ID. Enter the name of a synchronization profile as it is defined in Synchronization Profile Maintenance (36.8.22.1). Only one profile can be associated with an export specification.

Publish on Table Update. Enter Yes to create export documents for the specified profile whenever a record in the associated table is updated. This is event-driven, automatic data exporting.

For batch-only data synchronization, enter No and use Synchronization Mass Export (36.8.22.8) to create export documents for a specified profile.

Note The following frame displays only when you are setting up export specifications for synchronizing data between databases. If the Application ID specified is a domain system ID, this frame is not needed.

Fig. 16.19
Export Specification Maintenance, Messaging and Miscellaneous Parameters



Precede Data with Tags. Enter Yes to embed document control tags in front of the actual data when using the stream or messaging APIs to send documents to external databases. Control tags are always included when sending documents to a file.

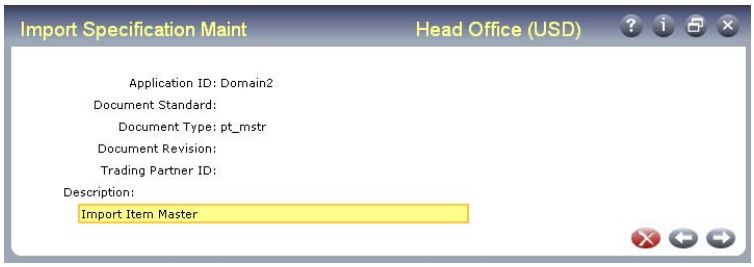
Access Code/Path. Specify the name of the domain in the external database to receive messages based on this export specification.

Note If you specified a destination list as the Application ID, any value you specify in this field is ignored. Q/LinQ uses the value specified in Register External Application for each application in the list.

Register Import Specifications

Use Import Specification Maintenance (36.8.1.3) to register each inbound synchronization document type. Accept the Q/LinQ defaults except where noted.

Fig. 16.20
Import Specification Maintenance (36.8.1.3)



Application ID. Enter the name of an external domain as it is defined in Register External Application (36.8.1.1) or the system ID of another domain within the current database defined in Q/LinQ Control (36.8.24).

Note Destination lists cannot be used with import documents.

Leave blank if you want this specification to apply to all documents of a certain type, standard, and revision without regard to the associated application.

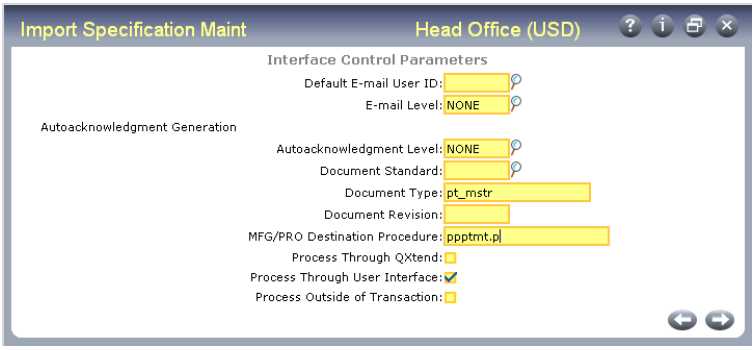
Document Standard. Enter a user-defined name such as sync_docs so that synchronization activity and documents can be easily segregated from other Q/LinQ documents and activities.

This field is validated against codes defined in Generalized Codes Maintenance for field esp_doc_std.

Document Type. Enter a user-defined name for the type of data that is being synchronized. Use names that reflect the type of data, such as pt_part for item numbers, to facilitate browsing, reporting, and tracking specific data elements.

Document Revision and Trading Partner ID can be left blank. If they are used, Document Standard, Document Type, Document Revision, and Trading Partner ID must be a unique combination of values.

Fig. 16.21
Import Specification Maintenance, Interface Control Parameters



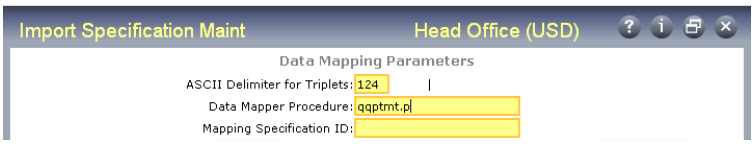
In the Interface Control Parameters frame, set up incoming processing and e-mail notification parameters.

Default E-mail User ID, E-mail Level. Enter data in these fields as needed or accept the defaults specified for the database in Register External Application.

Application Destination Procedure. Enter the name of the application program to call to process the data from the imported synchronization document as an application transaction. See Table 16.3 on page 276 for a list of programs.

Process Through User Interface. Enter Yes to invoke the destination procedure through the CIM Interface. Yes is required for synchronization.

Fig. 16.22
Import Specification Maintenance, Data Mapping Parameters



In the Data Mapping Parameters frame, specify the data mapping procedure from Table 16.3. Select the procedure that represents both the destination procedure and the table for the imported data type.

Table 16.3
Synchronization Data Mapping Programs

Destination Procedure	Supported Tables	Mapping Procedure Name
adcsmt.p	ad_mstr	qqadmp.p
	cm_mstr	qqcmmp.p
adlsmt.p	ls_mstr	qqlsmp.p

Table 16.3 — *Synchronization Data Mapping Programs* — (Page 1 of 2)

Destination Procedure	Supported Tables	Mapping Procedure Name
adrtmt.p	ad_mstr	qqadmp.p
	ls_mstr	qqlsmp.p
adstmt.p	ad_mstr	qqadmp.p
	cm_mstr	qqcmmp.p
advnmt.p	vd_mstr	qqvdmp.p
	ad_mstr	qqadmp.p
	ls_mstr	qqlsmp.p
bmmamt.p	bom_mstr	qqbommp.p
bmpsmt.p	ps_mstr	qqpsmt.p
csmsmt.p	cs_mstr	qqcsmp.p
fcfsmt01.p	fcs_sum	qqfcsmp.p
glenmt.p	en_mstr	qqenmp.p
glacmt.p	ac_mstr	qqacmp.p
glcalmt.p	glc_cal	qqglcmp.p
	glcd_det	qqglcdmp.p
glsbmt.p	sb_mstr	qqsbmp.p
	cr_det	qqcrmp.p
glccmt.p	cc_mstr	qqccmp.p
	cr_det	qqcrmp.p
gpcmmt.p	cd_det	qqcdmp.p
icsimt.p	si_mstr	qqsimp.p
icstmt.p	is_mstr isd_det	qqismp.p qqisdmp.p
mccuacmt.p	acdf_mstr	qqacdfmp.p
mccumt.p	cu_mstr	qqcump.p
mcexrmt.p	exr_rate	qqexrmt.p
mgcodemt.p	code_mstr	qqcodemp.p
ppacln.p	anl_det	qqanlmp.p
ppacmt.p	an_mstr	qqanmp.p
ppacrl.p	ans_det	qqansmp.p
ppcpmt.p	cp_mstr	qqcpmp.p
ppplmt.p	pl_mstr	qqplmp.p
pppcmt.p	pc_mstr	qqpcmp.p
pppimpt.p	pi_mstr	qqpimp.p
	pid_det	qqpidmp.p
ppptmt.p	pt_mstr	qqptmp.p
pppummt.p	um_mstr	qqummp.p
ppvpmt.p	vp_mstr	qqvpmp.p
rwdpmt.p	dpt_mstr	qqdptmp.p
rwromt.p	ro_det	qqromp.p
rwwcmt.p	wc_mstr	qqwcmp.p

Table 16.3 — *Synchronization Data Mapping Programs* — (Page 2 of 2)

Copy Import and Export Specifications

In a multi-domain environment that requires extensive sharing of master data, updates from each domain may need to be propagated to many other domains. Q/LinQ requires similar setup information for each destination domain in each source domain.

To facilitate the creation of similar specifications, you can use Export/Import Spec Copy (36.8.1.4) to copy an export or import specification, updating or creating the destination record.

This program can help you streamline the creation of similar specifications. For example, you can create template records and then make multiple copies of them. This minimizes the number of changes needed to complete setup activities.

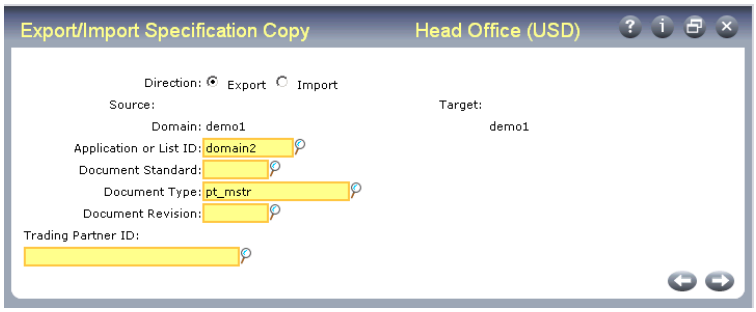
This function is especially important in a database that includes multiple domains that need to share master data. To support this synchronization, some setup information for other domains must be maintained in each domain that is going to share data. You can create one source import or export specification for each table to be synchronized and then copy it to the other related domains.

You must first create the source specifications using Export Specification Maintenance and Import Specification Maintenance.

You can select a specification from any domain as the source of the copy. The target specification is created in the user's current working domain. In a multiple-database environment, the source domain can exist in another database. The system automatically switches to that domain to find the source record.

If the destination specification exists, the system displays a warning and prompts you to continue. You can overwrite the existing record with values from the source specification.

Fig. 16.23
Import/Export Specification Copy (36.8.1.4)



Choose either Import or Export to indicate the type of specification record to be copied.

Domain. Enter the code identifying the domain associated with the source record to be copied.

You can choose any domain in this database or a connected, remote database. The destination domain defaults to your current working domain and cannot be changed.

Application ID. Enter the ID of the application associated with the source specification to be copied.

This can be the ID of an application registered in Register External Application or the Q/LinQ system ID of a domain as specified in
Q/LinQ Control. If you are copying an export specification, it can also be a destination list created in Destination List Maintenance.

To. Enter the ID of the application associated with the specification to be created or updated in the destination domain.

Document Standard, Document Type, Document Revision, Trading Partner ID. Enter values in these fields if required to identify the source specification.

To. Enter values for the destination specification if required. These can be the same as or different than the source specification.

Viewing Specification Lists

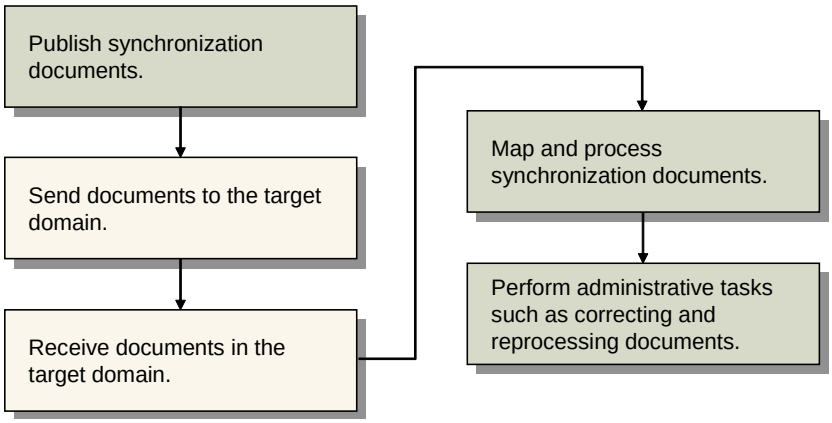
Use Export/Import Specification List (36.8.1.5) to view import or export specifications for a specified document standard, type, and revision combination. You can view records for a specific domain or for all domains. This lets you quickly see the subscription list of senders or receivers associated with a type of document.

Processing Synchronization Documents

After setup tasks are complete, you can begin synchronizing data among multiple, connected databases or among domains within a database.

Figure 16.24 illustrates a typical work flow for processing synchronization documents.

Fig. 16.24
Synchronization Processing Flow



When you are synchronizing data among domains in a single database using intra-database forwarding, the processing is simpler:

1. Documents are published directly to the import queue of the other domain.
2. Documents are mapped according to the import specification in the destination domain and processed to update the relevant tables.

When you are synchronizing data using the stream or messaging APIs between a source domain and domains in the same or connected databases, you must:

1 Publish synchronization documents.

2 Send the documents to Q/LinQ in the remote database.

3 Receive the documents in the remote database.

4 Map documents according to the import specification in the destination domain and process them to update the relevant tables.

Note When you are using the Q/LinQ-to-Q/LinQ adapter (`qqqq2qq.p`), the send and receive step are combined into one since this adapter can directly update the other database without going through a middleware product.

Publishing Documents

Synchronization documents are published automatically through schema triggers or manually through mass export programs.

Synchronization Triggers

With automatic publication, a synchronization document is published each time the table specified in a synchronization profile is updated. This is determined by the setting of Publish on Table Update in Export Specification Maintenance (36.8.1.2).

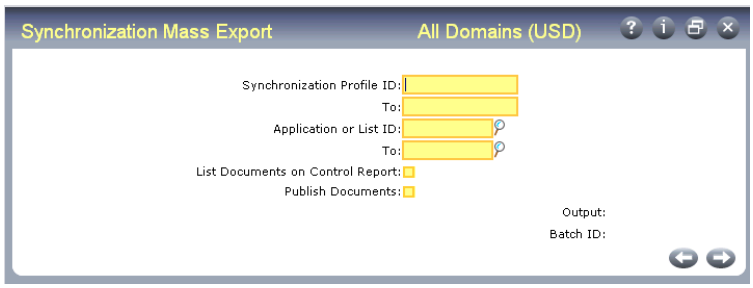
See “Publish on Table Update” on page 274.

Mass Document Export

Manual publication is a batch approach to creating synchronization documents. This approach is particularly useful for publishing large numbers of records during initial synchronization of multiple domains or databases.

Use Synchronization Mass Export (36.8.22.8) to publish the documents associated with a selected profile for selected destinations.

Fig. 16.25
Synchronization Mass Export (36.8.22.8)



Synchronization Profile ID and To. Enter a range of profile names. For a record from a table specified in a profile to be included in the mass export, any field that is marked as Required for Add must have a value.

Application or Destination List ID and To. Enter a range of destinations to receive the mass-exported documents. When synchronizing in batch between domains in a single database, enter a range of system IDs.

List Documents on Control Report. Enter Yes to include control information about each document on the control report. Enter No to have the report list only the number of document groups selected, processed, sent, containing errors, or skipped.

Control information includes:

- Document ID, group ID, application ID
- Published standard of the document that defines its structure and content after mapping (for example, ANSI X12 or OAGIS)
- Type of document (for example, 850 for ANSI X12 or SYNC SALESORDER for OAGIS BOD)
- Trading partner ID
- Processing stage (published, mapped, sent, or acknowledged)
- Error status (success, warning, failure)

Publish Documents. Enter Yes to publish the documents; enter No to generate the report only.

Sending and Receiving Documents

Important These steps are required only when synchronizing data using the messaging or stream API. When you synchronize data between domains in a single database using intra-database forwarding, Q/LinQ publishes the data directly to the import queue of the destination domain. Only the mapping and processing steps are required (see page 282).

In addition, if you use the Q/LinQ adapter (`qqqq2qq.p`) for between database communication, you only need to complete the send or the receive step, not both. This is the recommended approach, since it simplifies setup and processing.

If you use the Q/LinQ communication APIs (stream, messaging) or ASCII files to move synchronization documents between databases, the exchanges can be initiated interactively or through batch scripts.

See these topics in the chapter on “Managing Documents” in *External Interface Guide: Q/LinQ* for details:

- Send Export Document Sessions
- Using Send Export Documents
- Importing from External Applications
- Receiving Import Documents

Exchanging Documents Through APIs

If you do not use the recommended method for exchanging synchronization documents (the Q/LinQ-to-Q/LinQ adapter), you can also exchange documents through the stream and messaging APIs. This approach requires more setup.

Important On Windows systems, the messaging API is limited to a single session. If you are using destination lists for sending documents on Windows systems, use the stream API method.

Note When using the messaging API, start the listener process first. It runs as a server waiting for a caller to make contact. See “Messaging API” on page 268.

Use Send Export Documents (36.8.7) to export documents to the destination database:

- For Send To, choose Application.
- In Application ID, enter the application ID for the source and destination domain combination as defined in Register External Application. The stream or messaging API parameters are also defined for the application ID in Register External Application.
- In the second frame, specify the ranges defining the documents to export.

Use Receive Import Documents (36.8.9) to import documents from the source database.

- For Import From, choose Application.
- In Application ID, enter the application ID for the source and destination domain combination.

Exchanging Documents Through Files

Exchanging synchronization documents through ASCII files is a manual process. There are no Q/LinQ mechanisms for automatically transferring files between locations (hosts, directories) or for continuously polling directories for files.

Note User-written polling procedures can be created outside of Q/LinQ or using the Q/LinQ stream API to read files continuously into Q/LinQ.

Use Send Export Documents (36.8.7) to export synchronization documents to ASCII files:

- For Send To, select File.
- In File Name, enter the output path.
- In the second frame, specify the application ID for the source and destination domain combination as defined in Register External Application, as well as the ranges defining the documents to export.

Use Receive Import Documents to load ASCII files into the import queue of the domain in the destination database:

- For Import From, select File.
- In Source File Name, enter the path to the ASCII file of synchronization documents.

Mapping and Processing Documents

Use Process Import Documents (36.8.10) to map the synchronization data from the received documents to the appropriate format and to update the destination domain.

In a database with multiple domains, you can use this function to process documents for one domain or for all domains by leaving the Domain field blank. Your current working domain is the default value. If you have centralized Q/LinQ administration, this approach reduces the number of Q/LinQ jobs that need to be started, monitored, and stopped.

Mapping and processing can be initiated interactively or through a batch script. See “Mapping and Processing Import Documents” in *External Interface Guide: Q/LinQ* for details.

To increase processing throughput, run multiple, concurrent sessions of Process Import Documents. It is best to have each session processing different document types and document ranges since concurrent sessions do not preserve the chronological order of synchronization actions.

Performing Q/LinQ Administration

Once Q/LinQ is set up and documents are being processes, Q/LinQ administrators may need to perform a number of routine administration tasks such as:

- Tracking documents
- Correcting and reprocessing documents
- Dumping documents to a text file
- Deleting documents
- Managing Q/LinQ sessions

Tracking Documents

To facilitate document tracking, create unique codes during setup:

- Use Register External Application (36.8.1.1) to specify a unique application ID for each pair of domains in this and an external database that will exchange documents. See page 273.
- Use Export Specification Maintenance (36.8.1.2) to specify a unique document standard or document type for each synchronization document. See page 275.

With these unique definitions in place, use the standard Q/LinQ tools to monitor and report the status of synchronization activity by database, domain, or document:

- Export/Import Document Query (36.8.16)
- Export/Import Document Report (36.8.17)

Both of these reports let you monitor activity in one or all domains. Your current working domain is the default, but you can specify any other domain or leave the Domain field blank to view activity in all domains.

For documents exported to destination lists, these reports display export log information for each destination.

See “Managing Documents” in *External Interface Guide: Q/LinQ*.

Correcting and Reprocessing Documents

Occasionally a receiving domain will fail to process a synchronization document as a transaction. This can occur, for example, when all prerequisite codes are not set up in the destination domain.

For CIM documents, use Debug CIM Document (36.8.11) to interactively process the document to more clearly identify the error source.

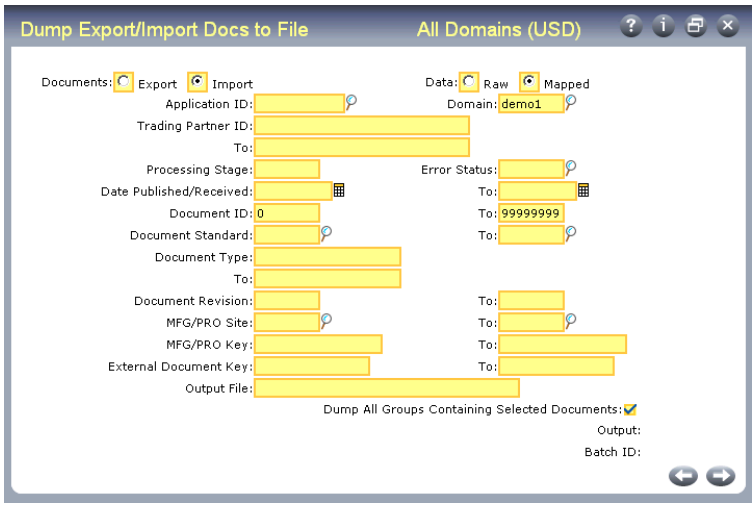
Note In extreme cases only, use Dump Export/Import Doc for Edit (36.8.13) to manually correct the data inside the destination domain and Reload Edited Export/Import Doc (36.8.14) to reload it for processing.

Dumping Documents to File

Use Dump Export/Import Docs to File (36.8.18) to transfer a range of documents to a text file. In a multiple-domain environment, you can dump documents for one or all domains. Your current working domain is the default, but you can specify any other domain or leave the Domain field blank to dump documents associated with all domains.

This function is most useful for copying existing documents, altering them, then loading the new documents into Q/LinQ using Receive Import Documents (36.8.9).

Fig. 16.26
Dump Export/Import Docs to File (36.8.18)



Warning Do not use this procedure to dump documents for transport to external applications. Since it is not a Q/LinQ export process, Q/LinQ cannot track the documents and does not record the dumped documents as sent in the log.

To export documents to other applications, use Send Export Documents (36.8.7). To dump documents to archives, use Export/Import Doc Delete/Archive (36.8.23).

You can dump the raw document data or the mapped data. Q/LinQ dumps raw data if mapped data is requested but not present.

The output is a text file of data lines wrapped in header lines. Identifier values in the header describe key document attributes, such as the document type and the source or destination application.

Deleting Documents

Since Q/LinQ does not automatically delete completed export or import documents, the number of synchronization documents grows quickly in high-volume environments. Use Export/Import Doc Delete/Archive (36.8.23) to remove completed documents from the export and import queues at least once a day. Cleaning up the queues reclaims storage space and enhances Q/LinQ performance. Export documents can be chosen by destination list as well as by application.

For documents exported to destination lists, Export/Import Doc Delete/Archive deletes export log information for each destination.

In a database with multiple domains, you can use this function to delete and archive documents for one domain or for all domains by leaving the Domain field blank. Your current working domain is the default value. If you have centralized Q/LinQ administration, this approach reduces the number of Q/LinQ jobs that need to be managed.

See “Deleting or Archiving Documents” in *External Interface Guide: Q/LinQ*.

Managing Sessions

Tracking Sessions

Use Interface Session Monitor (36.8.5) to pause, resume, cancel, or view the status of a session.

For sessions exporting to destination lists, the connections to all databases in the list are controlled through one primary session record. Only the primary session record can be updated directly to pause, resume, cancel, or delete the session. Each discrete connection has its own session master record in the database; these are displayed for inquiry only.

See “Monitoring Q/LinQ Sessions” in *External Interface Guide: Q/LinQ*.

Recovering from Communication Errors

When Q/LinQ encounters a communication error when sending a document to a single application or database, Q/LinQ attempts to reconnect to the destination database and to resend the document.

Note These kinds of errors would never occur when synchronizing data among domains in a single database.

If a communication error occurs when sending a document to a destination list, Q/LinQ logs the error and goes on to the next destination database rather than attempting to reconnect and resend to the destination database having communication problems. In this way, an error affecting one destination does not delay the delivery of a document to the other destinations.

You can take advantage of the reconnect and resend functionality by publishing a document to a destination list and then sending the document to each of the destination databases one at a time in separate Q/LinQ sessions.

Use the Interface Session Monitor to pause, resume, or cancel the export session at any time. If the session is paused or canceled, a message displays in the Send Export Documents window. If a session is canceled (either through Interface Session Monitor, pressing End, or a system failure), manually delete the session record to release the external application so that other Q/LinQ sessions can connect to it. Delete session records using the delete option in Interface Session Monitor.

Restarting Sessions

After severed communication caused by network problems or user interrupts (Ctrl+C) is restored, the system administrator can restart the databases. Any documents that were being sent or received at the time of the communication loss remain intact in their source database. Each document has an error status indicating that the sending process was interrupted. The documents can be resent once communication is restored and the databases are restarted.

See “Starting and Restarting Q/LinQ” in *External Interface Guide: Q/LinQ*.

Index

E

F

G

H

I

J

K

S