Industry-specific

QAD SOLUTIONS

Manufacturing Applications

MFG/PRO eB2 User Guide Volume 9
Manager Functions

System Constants
Users and Security
System Interface
Multiple Databases
Printers and Batch Processing
CIM Interface
Database Management
Reports and Utilities
System Cross-Reference
Application Server
User Interface Management

78-0565A
MFG/PRO eB2
September 2002

78-0565A

QAD Inc.
6450 Via Real
Carpinteria, California 93013
Phone (805) 684-6614
Fax (805) 684-1890
http://www.qad.com

Contents

About This Guide

This guide covers the system administration programs within MFG/PRO. Most of these programs are on the Manager Functions menu (36).

Other MFG/PRO Documentation

- For an overview of new features and software updates, see the *Release Bulletin*.
- For software installation instructions, refer to the appropriate installation guide for your system.
- For conversion information, refer to the *Conversion Guide*.
- For instructions on navigating and using the QAD Desktop interface, see *User Guide: QAD Desktop.*
- For instructions on navigating the MFG/PRO Windows and character environments, refer to *User Guide Volume 1: Introduction*.
- For information on using MFG/PRO, refer to the *User Guides*.
- For technical details, refer to *Entity Diagrams* and *Database Definitions.*
- For information on using features that let MFG/PRO work with external applications, see the *External Interface Guides*. Each book in this set describes a separate interface such as the Warehousing application program interface (API) and Q/LinQ, the tool set for building and using data exchange tools.
- To view documents online in PDF format, see the *Documents on CD* and *Supplemental Documents on CD*. The CD-ROM media includes complete instructions for loading the documents on a Windows network server and making them accessible to client computers.

  Note MFG/PRO installation guides are not included on a CD. Printed copies are packaged with your software. Electronic copies of the latest versions are available on the QAD Web site.

Online Help

MFG/PRO has an extensive online help system. Help is available for most fields found on a screen. Procedure help is available for most programs that update the database. Most inquiries, reports, and browses do not have procedure help.

For information on using the help system in the different MFG/PRO environments, refer to *User Guide Volume 1: Introduction* and *User Guide: QAD Desktop*.

QAD Web Site

QAD's Web site provides a wide variety of information about the company and its products. You can access the Web site at:

http://www.qad.com

For MFG/PRO users with a QAD Web account, product documentation is available for viewing or downloading at:

http://support.qad.com/documentation/

You can register for a QAD Web account by accessing the Web site and clicking the Accounts link at the top of the screen. Your customer ID number is required. Access to certain areas is dependent on the type of agreement you have with QAD.

Most user documentation is available in two formats:

- Portable document format (PDF). PDF files can be downloaded from the QAD Web site to your computer. You can view them with the free Adobe Acrobat Reader. A link for downloading this program is also available on the QAD Web site.
- HTML. You can view user documentation through your Web browser. The documents include search tools for easily locating topics of interest.

Features also include an online solution database to help MFG/PRO users answer questions about setting up and using the product. Additionally, the QAD Web site has information about training classes and other services that can help you learn about MFG/PRO.

Conventions

MFG/PRO is available in several interfaces: Desktop (Web browser), Windows, and character. To standardize presentation, the documentation uses the following conventions:

- MFG/PRO screen captures show the Desktop interface.
- References to keyboard commands are generic. For example, choose Go refers to:
  - The forward arrow in Desktop
  - F2 in the Windows interface
  - F1 in the character interface

  In the character and Windows interfaces, the Progress status line at the bottom of a program window lists the main UI-specific keyboard commands used in that program. In Desktop, alternate commands are listed in the right-click context menu.

  For complete keyboard command summaries for each MFG/PRO interface, refer to the appropriate chapters of *User Guide Volume 1: Introduction* and *User Guide: QAD Desktop*.

This document uses the text or typographic conventions listed in the following table.

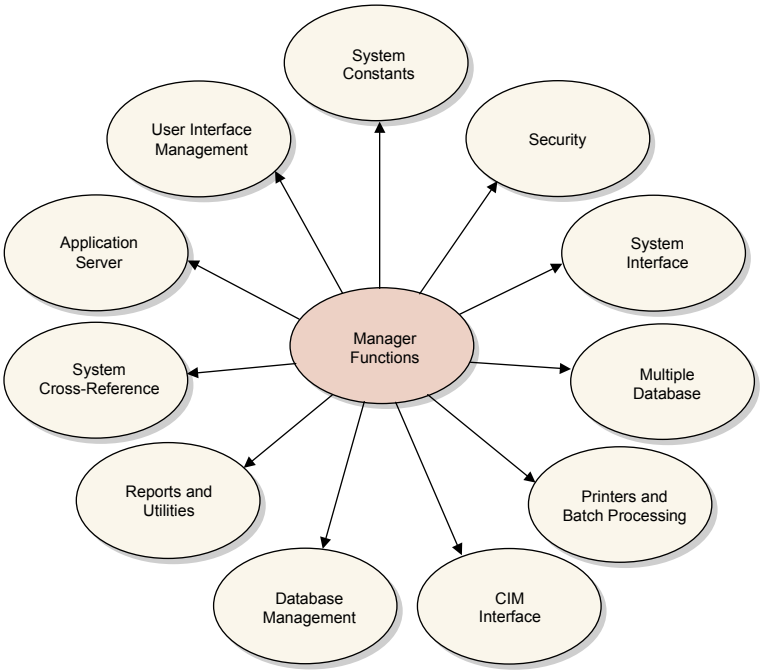
If you see:	It means:
`monospaced text`	A command or file name.
`italicized monospaced text`	A variable name for a value you enter as part of an operating system command; for example, *`YourCDROMDir`*.
`indented command line`	A long command that you enter as one line, although it appears in the text as two lines.
Note	Alerts the reader to exceptions or special conditions.
Important	Alerts the reader to critical information.
Warning	Used in situations where you can overwrite or corrupt data, unless you follow the instructions.

Chapter 1

Introduction to Manager Functions

Manager Functions includes tasks typically performed by system administrators within MFG/PRO. Most functions located on the Manager Functions menu (36) are discussed in this volume.

Fig. 1.1
Manager Functions



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

- System/Account Control (36.1) affects processes throughout MFG/PRO. 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 Volume 4A: Financials*.
- The functions on the Desktop Security Menu (36.3.21) are discussed in the *User Guide: QAD Desktop*.
- Configured Messaging (36.4) applies only to scheduled orders and is discussed in *User Guide Volume 7: Release Management.*

- External Interfaces (36.5), Q/LinQ (36.8), and the Logistics API (36.5.7) are discussed in various *External Interface Guides*.
- User interface functions found on the Desktop Menu (36.20.10) are discussed in the *User Guide: QAD Desktop* and *Installation Guide: QAD Desktop.*

This volume 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.

System Constants

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

See Chapter 2, "System Constants," for details.

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.

Security

MFG/PRO requires that a user be defined with a valid ID and password before they can log in. In addition, the system offers several types of security, including menu, field, entity, site, account, and inventory movement code. You can implement these levels by user ID or user group.

See Chapter 3, "Users and Security," for details.

System Interface

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.

See Chapter 4, "System Interface," for details.

Multiple Database

See Chapter 5, “Multiple Databases,” for details.

The Multiple Database menu includes programs for connecting to and setting up multiple databases. This chapter also covers the system requirements for operating with multiple databases.

Printers and Batch Processing

See Chapter 6, “Printers and Batch Processing,” for details.

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

See Chapter 7, “CIM Interface,” for details.

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

Database Management

See Chapter 8, “Database Management,” for details.

MFG/PRO 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

See Chapter 9, “Reports and Utilities,” for details.

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

System Cross-Reference

See Chapter 10, “System Cross-Reference,” for details.

The system cross-reference programs display information about field, program, and table relationships in your database. If you customize MFG/PRO, this is an essential set of programs.

Application Server

MFG/PRO can use a Progress application server (AppServer) to run applications remotely. The AppServer must be defined in MFG/PRO to make it available.

See Chapter 11, “Application Server,” for details.

User Interface Management

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.

See Chapter 12, “User Interface Management,” for details.

Chapter 2

System Constants

The programs on the System Constants menu control calendars and codes used throughout the system.

Overview

System constants provide basic data used throughout MFG/PRO. Table 2.1 lists available functions.

Table 2.1
System 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. 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.

Tip
At least one calendar must exist.

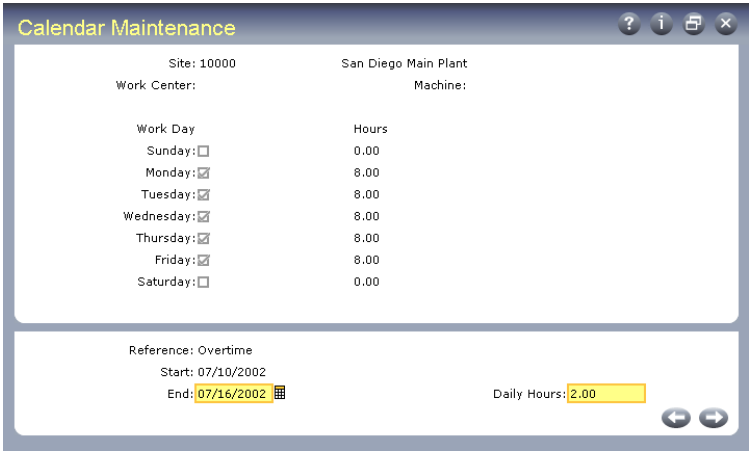


Fig. 2.1
Calendar Maintenance (36.2.5)

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. 2.2
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

MFG/PRO 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).

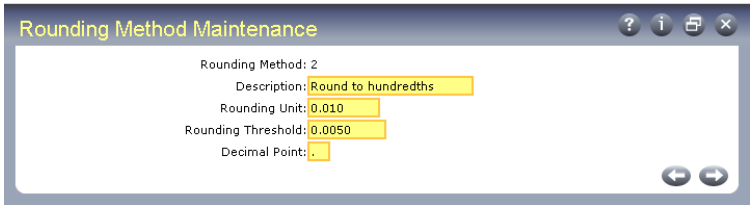


Fig. 2.3
Rounding Method Maintenance (36.2.9)

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.

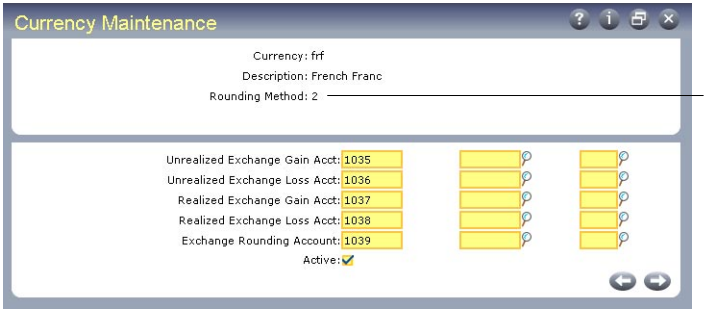
Rounding Threshold. Enter the number at which monetary values are rounded up. This number must be less than the number entered for the rounding unit.

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. 2.4
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 MFG/PRO 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 section of MFG/PRO, the implementation team should determine which fields should have generalized codes and lookups.

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 18.

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, the field name displays at the bottom of the screen when you press Ctrl+F with your cursor in the field. If the message 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 Desktop interface.

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 2.5 illustrates Generalized Codes Maintenance (36.2.13).

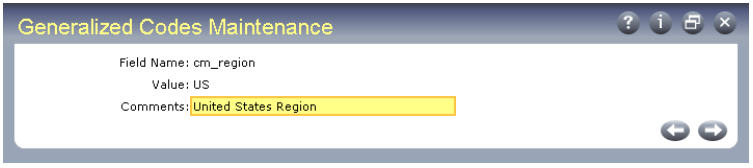


Fig. 2.5
Generalized Codes Maintenance (36.2.13)

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.

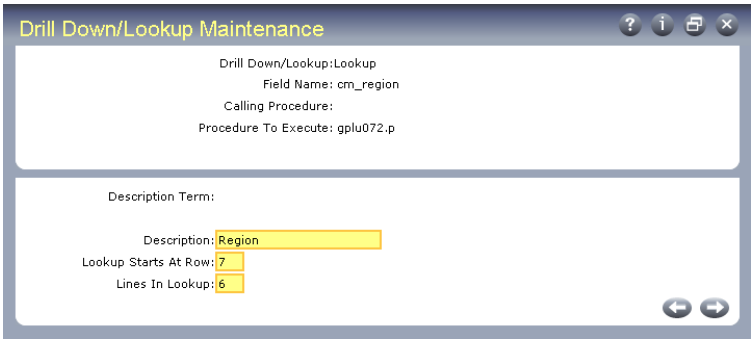
Adding a Lookup

See “Maintaining Drill Downs and Lookups” on page 178.

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.

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. 2.6
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 versions of MFG/PRO that update the data dictionary are installed.

Using Reason Codes

Reason codes are used in 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 PRO/PLUS modules, such as WIP Lot Trace and Shipment Performance. Add other custom uses as needed.

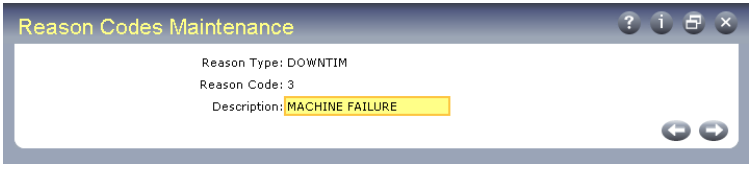


Fig. 2.7 Reason Codes Maintenance (36.2.17)

- 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 29.

- 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 PRO/PLUS 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 Volume 11: PRO/PLUS*.

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 2.8 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 (06:15:02).

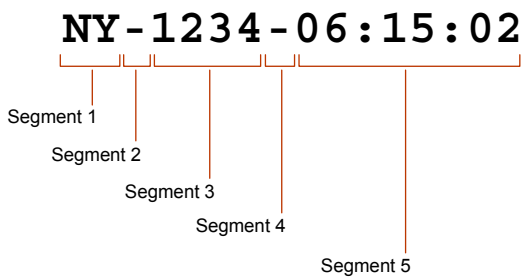


Fig. 2.8
Example Sequence Number

Table 2.2 describes the three segment types.

Table 2.2
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, 06:15:02 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 2.9 illustrates the life cycle of a sequence.

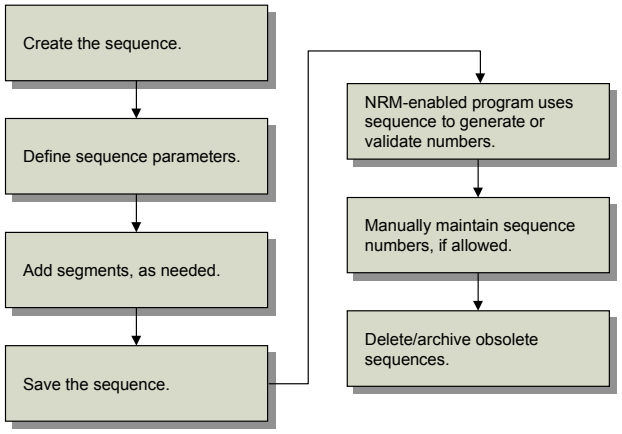


Fig. 2.9
Sequence Life Cycle

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

Tip
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 in MFG/PRO

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

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 Volume 4B: Financials*.

General Ledger Daybooks

See *User Guide Volume 4A: Financials*.

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.

Logistics Accounting

See *User Guide Volume 6: Master Data*.

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.

Shipping

See *User Guide Volume 2A: Distribution*.

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.

WIP Lot Trace

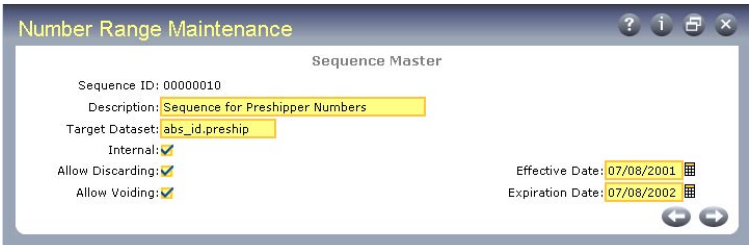
See *User Guide Volume 11: PRO/PLUS*.

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.

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. 2.10
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.

Tip
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

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.

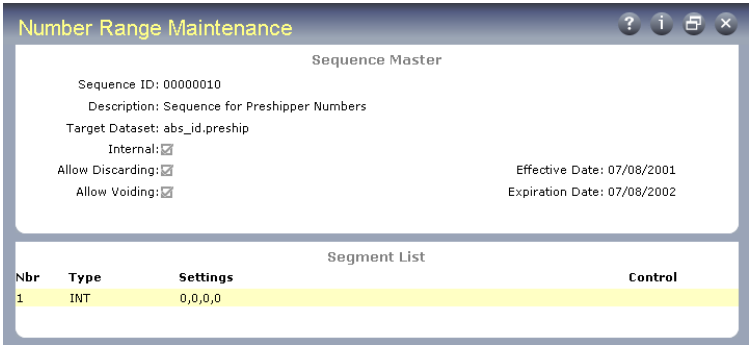


Fig. 2.11 Number Range Maintenance (36.2.21.1), Segment List Frame

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. 2.12 Fixed Segment Editor

Integer Segment Editor

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

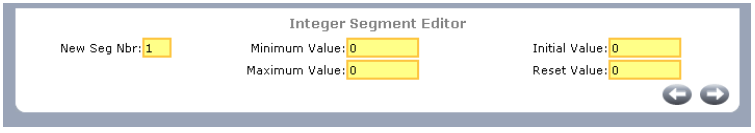


Fig. 2.13 Integer Segment Editor

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.

Tip 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. 2.14 Date Segment Editor

Fiscal Segment Editor

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

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.

Fig. 2.15
Fiscal Segment Editor

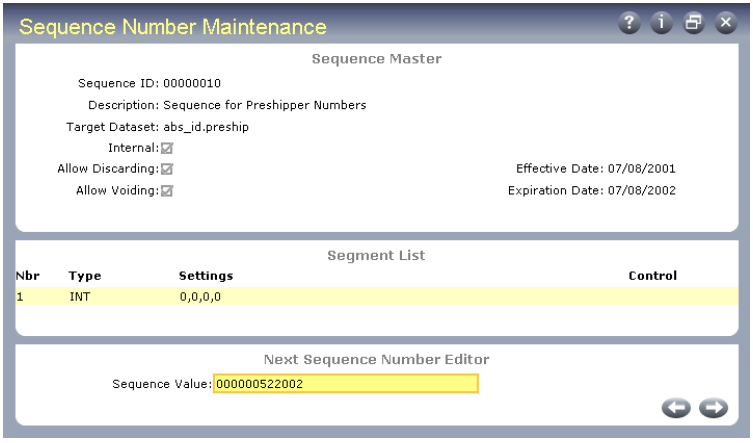
Setting Sequence Values

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 sequence value you enter is validated against the segment order and settings.

Fig. 2.16
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.

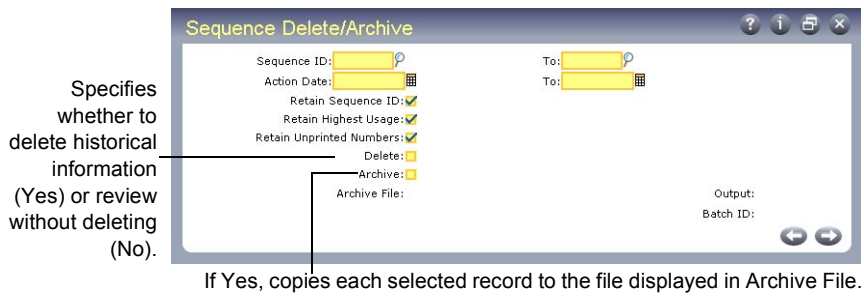


Fig. 2.17 Sequence Delete/ Archive (36.2.21.23)

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.

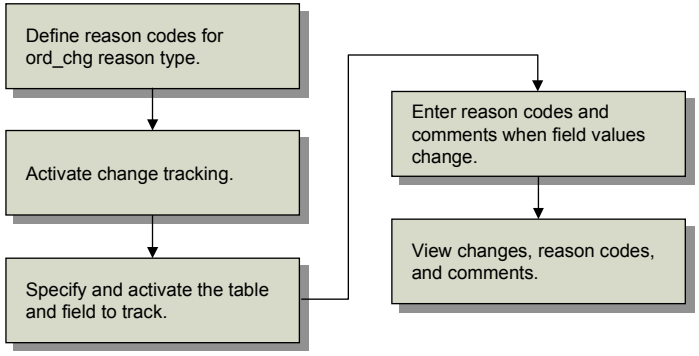
- Allow users to enter a reason code and comments when the value of a marked field changes.
- Print the changes, reason codes that explain the changes, and any associated comments on a Booking Transaction Report (7.15.14).

See *User Guide: Volume 2A: 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 2.18 illustrates the basic change tracking implementation flow.

Fig. 2.18 Change Tracking Implementation Flow



Defining Change Tracking Reason Codes

See "Using Reason Codes" on page 19.

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.

Activating Change Tracking

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

See *User Guide Volume 2A: Distribution*.

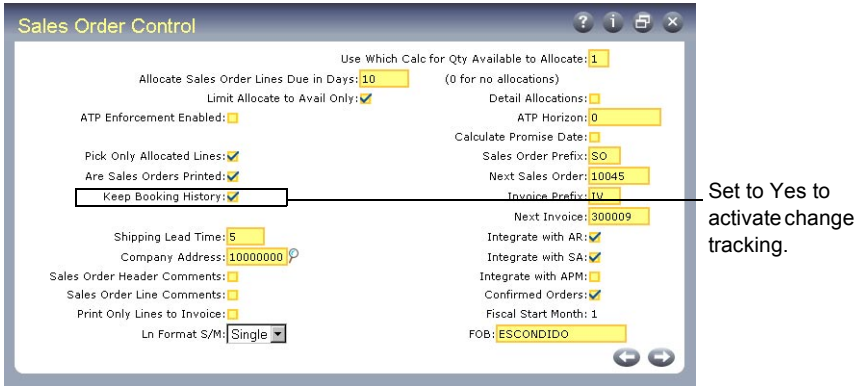


Fig. 2.19
Sales Order Control (7.1.24)

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.

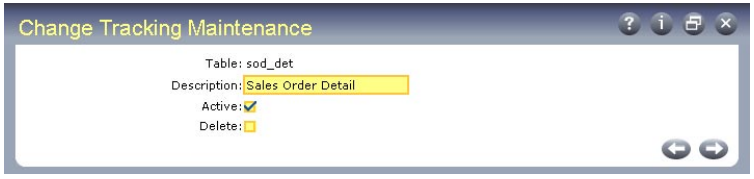


Fig. 2.20
Change Tracking Maintenance (36.2.22)

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. 2.21
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 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).

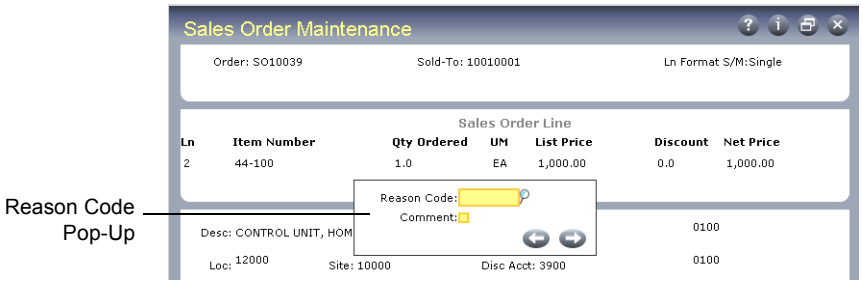


Fig. 2.22
Reason Code Pop-Up in Sales Order Maintenance (7.1.1)

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

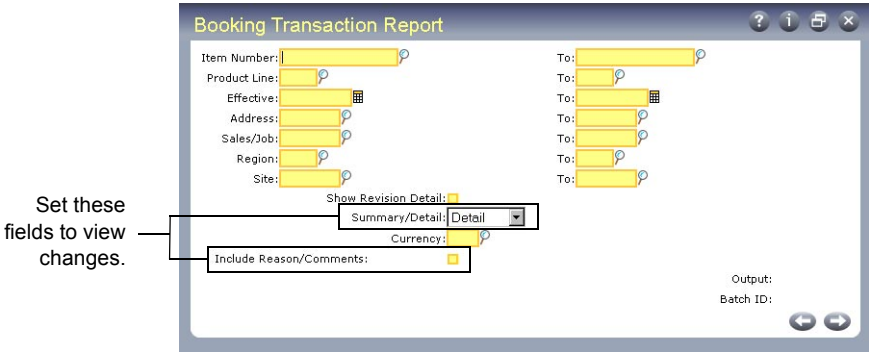


Fig. 2.23
Booking Transaction Report (7.15.14)

Chapter 3

Users and Security

This chapter describes how to set up users and manage different kinds of security.

Security Overview

MFG/PRO provides options for security on several levels.

- Database security determines whether a user can log into a database. This level of security is always active and requires that users specify a valid user ID and password before they can log in.
- Menu security (36.3.1) limits access to menus and menu functions.
- Field security (36.3.4) limits who can update specific fields.
- General ledger (GL) account security (36.3.9) restricts access to GL accounts.
- Inventory movement security (36.3.11) enables you to grant or deny group members access to shippers and other transactions using specific movement codes at a site.
- Entity security (36.3.13) limits who can create GL transactions for a particular entity.
- Site security (36.3.16) limits who can create inventory transactions at secured sites.

See *User Guide: QAD Desktop* for details.

- Desktop security (36.3.21) lets you restrict who can log into a database using QAD's Desktop interface. It also provides a detailed and focused way of controlling access to data when browses and reports are generated from the Desktop.

See page 156 for details.

- Master table audits (36.17 menu) show when and by whom changes were made to key master tables.

Figure 3.1 illustrates how several different kinds of security can operate at the same time with the same user.

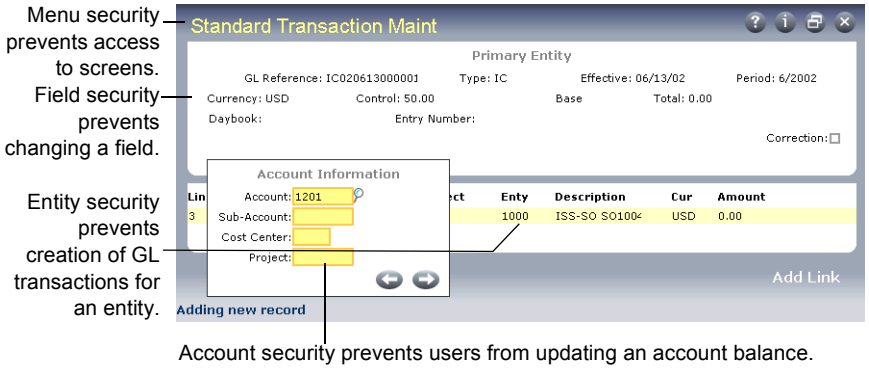


Fig. 3.1
Types of Security

By default, only log-in security is defined in MFG/PRO. Once you set up explicit permission for one user for entities, fields, or menus, all other users are excluded. For this reason, you should have a comprehensive security plan before beginning to set up security records.

All security functions use the same internal strategy. The security maintenance function creates a record, which pairs a field or function and user IDs or user groups.

- For entity and field security, specify user IDs.
- For menu security, site security, inventory movement code security, and GL account security, specify any combination of user IDs or user group names.

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 limits the user's actions accordingly.

Since the strings passed to a security function vary, the term *session ID* refers to any string passed to a security function that identifies the user. The session ID can be a user ID, a user group, or a combination of these.

Table 3.1 lists the programs used for system security.

Table 3.1
System Security Menu (36.3)

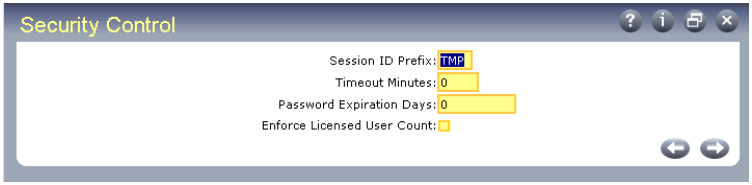
Number	Description	Program
36.3.1	Menu Security Maintenance	mgpwmt.p
36.3.2	Menu Security Change	mgpwcg.p
36.3.4	Field Security Maintenance	mgflpwmt.p

Number	Description	Program
36.3.5	Activated Field Security Report	mgflpwrp.p
36.3.6	Dictionary Field Security Report	mgfldcrp.p
36.3.7	Field Security by Group	mgflgpmt.p
36.3.9	GL Account Security Maintenance	mgacsmt.p
36.3.10	GL Account Security Report	mgacsrp.p
36.3.11	Inventory Movement Code Security	sosimt.p
36.3.11	Inventory Movement Code Browse	gpbr502.p
36.3.13	Entity Security Maintenance	glsecmt.p
36.3.14	Entity Security Inquiry	glseciq.p
36.3.15	Entity Security Report	glsecrp.p
36.3.16	Site Security Maintenance	clsismt.p
36.3.16	Site Security Report	clsisrp.p
36.3.18	User Maintenance	mgurmt.p
36.3.19	User Inquiry	mguriq.p
36.3.20	User Password Maintenance	mgurmtp.p
36.3.20	User Access by Application Inquiry	lvusriq.p
36.3.24	Security Control	mgurpmmt.p

Setting Up Security Control

Use Security Control (36.3.24) to establish basic security parameters for a database.

Fig. 3.2
Security Control (36.3.24)



Session ID Prefix. Enter a prefix for temporary system-generated work files. These are created in the directory where the user started MFG/PRO. 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.

Password Expiration Days. Set a value to require users to change their passwords after this number of days. Users who have not changed their passwords within the number of days specified are prompted to enter a new password at the sign-on screen. First, they must enter their correct old password, followed by a new password, which must be different from the old password and cannot be blank.

Date Password Last Changed in User Maintenance (36.3.18) displays the date the user last modified the password.

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

See "Registering Licenses" on page 136 for details.

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:

See Table 8.4 on page 140.

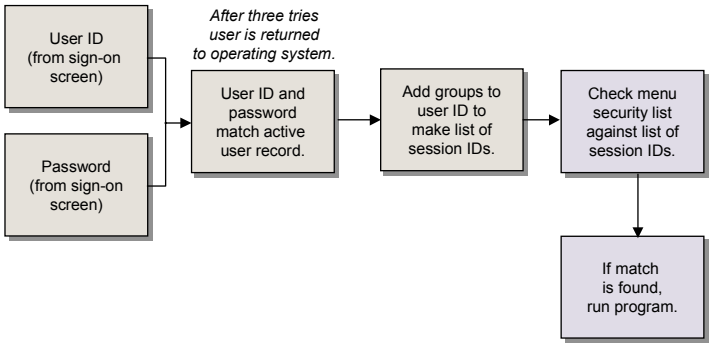
- In User Maintenance (36.3.18) when you attempt to add users or assign them to applications
- In User Maintenance when you attempt to change access location
- In License Registration (36.16.10.1) when you assign users to applications
- During user log-in to the system
- When users attempt to use separately licensed applications or nonregistered applications

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

Basic Login Security

A user must enter both a user ID and a password to log in to the database. If the user enters an invalid combination, the system prompts for input two more times. After three failures, the user is returned to the operating system. Figure 3.3 illustrates how this process occurs during log-in.

Fig. 3.3
Log-In Validation



This type of log-in security lets you:

- Effectively separate MFG/PRO security from the operating system security. The user ID in MFG/PRO does not have to be the same as the user ID referenced by UNIX or NT.
- 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 password to access MFG/PRO presents an additional barrier to an unauthorized user.

Important You should not ordinarily set up *root* as a user in this system. In Progress, the root ID has access to all functions with no security constraints. If root is defined as an MFG/PRO user, anyone with access to the root user ID at the operating system level can get access to any information in the MFG/PRO databases.

Progress Security

If necessary, you can use Progress utilities to place additional control over how a user's ID is set. When Progress security is in place, errors are generated before the MFG/PRO log-in screen displays.

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.

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.

Saving the User ID

For users of the Windows interface, MFG/PRO provides an option to save the user ID for the next login.

If you need to log in to MFG/PRO under a different user ID, uncheck the Save User ID for next login setting to prevent the system from resetting your default login.

Defining Users

You define users by assigning a unique user ID in User Maintenance (36.3.18). Each program is always passed the user's ID, any group names associated with the user, and access information associated with the user. After you create the ID for a user, you specify other identifying information and preferences.

To log in to MFG/PRO, each user must specify a unique user ID and a password that matches the user ID and password that you define here. Other user data is referenced throughout MFG/PRO and may be required for reasons other than security.

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. Inactive users can be archived during a QAD audit. 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.

Interaction with Licensing

There are three license types for MFG/PRO: named user, concurrent session, and location. Concurrent session licensing is checked at log-in. Both named user and location licensing is verified in User Maintenance.

See "Registering Licenses" on page 136.

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.

Location licenses stipulate a predefined number of locations for specific applications. System administrators define a user's access location in User Maintenance. When system administrators assign users to an application, the system counts the number of locations and compares the number against the predefined limit for the license type.

If the number of active users or locations 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.

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:

See “Setting Up Security Control” on page 38.

- Errors or warnings display in User Maintenance.
- System administrators can create new users when the number of existing users exceeds the licensed number.
- System administrators can assign applications to users for more locations than assigned in the license agreement.
- 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 MFG/PRO application. If a user ID is obsolete, you should inactivate access to all registered applications.

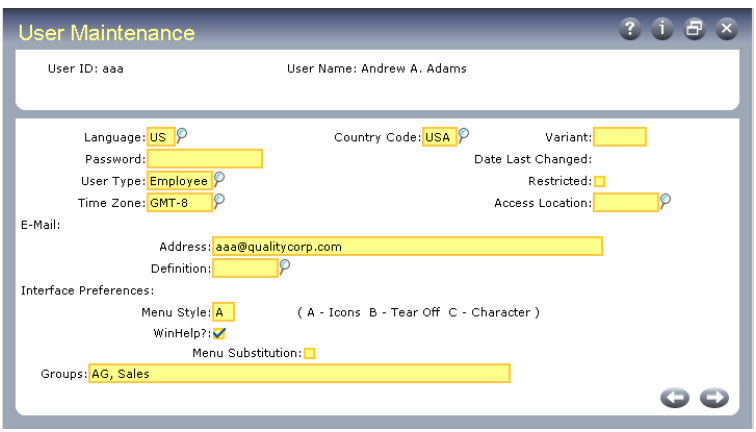


Fig. 3.4
User Maintenance (36.3.18)

Specifying Authorization and Access

Use the following fields to set up authorization and specify access for each user:

Password. Enter the password associated with this user ID. The user must enter the password to log in to MFG/PRO. If you do not want users to enter passwords at log-in, leave this field blank.

Note To change an existing password to blank, press the spacebar in the Password field and then press the backspace key. This action lets the system know that you want to change the password value.

See "Password Expiration Days" on page 39.

Passwords expire based on the value of Password Expiration Days in Security Control. If you want users to change their own passwords, give them access to User Password Maintenance (36.3.20).

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.

See "License Types" on page 137.

The system uses the access location when verifying user access to applications with the location license type. Location licenses specify a predefined number of locations for specific applications. The system validates location count when you change the access location of a user or when you assign applications to users in User Maintenance or License Registration (36.16.10.1). A message displays if the number of locations exceeds the amount specified in the license agreement.

Access location codes must be defined in Generalized Codes Maintenance (36.2.13) for field usr_access_loc. MFG/PRO 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.

Information Process and Display

You can ensure that MFG/PRO data are correctly displayed and processed for a given user, regardless of the user's language or location by specifying the following fields:

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.

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

See *User Guide Volume 6: Master Data.*

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 MFG/PRO 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 MFG/PRO 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.

See "Setting Up Multiple Time Zones" on page 149.

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 System/Account Control (36.1).

Specifying E-mail Addresses

See "Building an E-Mail System Interface" on page 74.

Associate a valid e-mail address and definition with each user who receives messages generated by MFG/PRO.

Setting Interface Preferences

Select interface preferences for individual users by specifying values for the following:

- Menu style. This only affects the menu style used in an MFG/PRO Windows session.

See page 47.

- Appearance of online help. This only affects the display of help in a Windows session. Select WinHelp? for Windows-style help files. If your system has been updated with custom help—set up in Field Help Maintenance (36.4.15)—you can view this additional help only when WinHelp is *not* selected.

See "Adding User Help" on page 73.

- Whether menu substitution is enabled or disabled. This only affects menus in Windows and character sessions. This does not affect the Desktop interface, where menu substitution is assumed to be always on.

See "Setting Up Menu Substitutions" on page 185.

Selecting a Menu Style

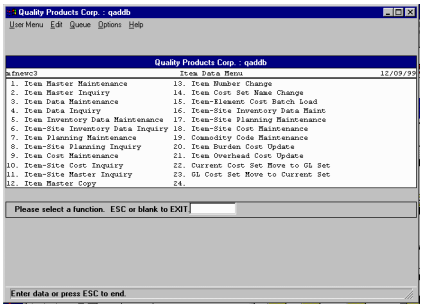
The Windows user interface includes three menu styles.

- 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.

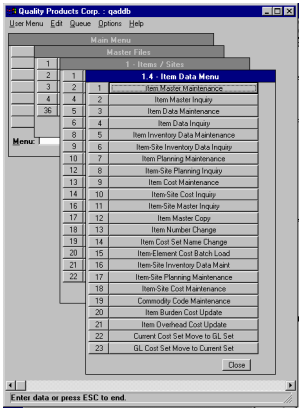
See *User Guide Volume 1: Introduction* for more information on menu styles and other elements of the user interface.

Fig. 3.5
Menu Style Options (GUI Interface Only)

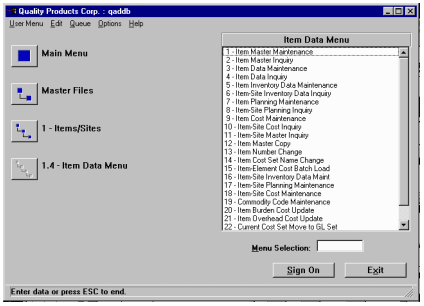
Character Menu



Tear-Off Menu



Icon Menu



Specifying User Groups

For an example, refer to “User Group Example” on page 52.

The groups a user belongs to can determine whether a user is given access to menus, fields, and sites. The system always passes the user’s ID and any group names associated with the user to each program.

Assigning users to groups makes setting up security much easier.

- Group names can contain letters, numbers, and some special characters. They cannot contain an asterisk (*), exclamation point (!), or comma (,).
- You can add multiple groups at one time. Separate group names with commas. Spaces are not necessary and are eliminated upon entry.
- You can specify a user ID when adding groups to allow access to individual users.
- You can add as many groups as needed to a user.
- Group names should be based on your company’s structure. Some examples follow:

- site100, site105, site110
- prod, plan, sales
- admin, supvsr, clrk

- The system does not validate group names when they are entered. Be careful to enter group names correctly for each user and ensure they match the groups specified in security functions.

Specifying Application Use

The Application List frame in User Maintenance lets you define the software applications that a user can access. MFG/PRO is listed as the default application. You can list additional software applications, then set Active to Yes or No for each application. The default is Yes.

The application name you enter under Application Name must be registered with MFG/PRO through License Registration (36.16.10.1).

See “Registering Licenses” on page 136.

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

If you deactivate MFG/PRO 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, time zone, access location, and access date.

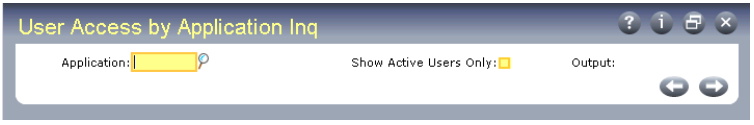


Fig. 3.6
User Access by Application Inquiry (36.3.22)

Using Menu Security

Menu security controls each user's access to programs. Use Menu Security Maintenance (36.3.1) to define the users or groups that have access to a menu function.

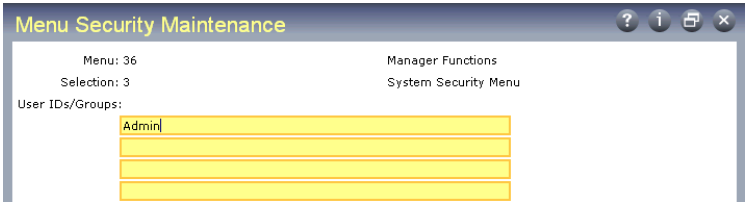


Fig. 3.7
Menu Security Maintenance (36.3.1)

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.

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.

Specifying Groups

Any number of session IDs, separated by commas, can be used, and both wildcards and exclusions are possible. At least one session ID, usually an asterisk, must appear for each menu item.

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

- Use the asterisk to give access to all users and groups. A blank operates the same way as an asterisk, allowing access to all users.
- The exclamation point restricts specific users by user ID, not by group. For example: `!user1,*` means all users except user1 have access to the program.
- 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,*
  ```

Table 3.2 lists some examples. Session IDs are not case-sensitive.

Table 3.2 Session ID Examples

String	Description
`*`	All users have access.
`!,*`	No users have access.
`admin,*`	All users whose session ID begins with admin have access.
`mary, manager`	Only users using the session IDs mary and manager have access.
`!jcd,*`	Everyone but the person whose session ID is 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` session ID would not be excluded. In general, because of this problem, avoid using any exclamation point after the first entry.

Effect of Menu Security

The effect of menu security varies according to the interface.

- In the character interface, users cannot see restricted menu items.
- In the Windows interface, a restricted menu item 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 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 (36.3.1) and specify selection 0 (zero), which represents the editor.

User Group Example

In this example of menu security for user groups, a company has three levels of access to accounts payable: one for clerks, one for managers, and one for the CFO.

Example Company A creates three groups: *Clerk*, *Manager*, and *CFO*. Helen, the AP Clerk, is added to the Clerk group. Don, the AP Manager, is added to the Manager and Clerk groups. Sara, the CFO, is added to all three groups. In this setup, Sara's session ID grants her entry to all the levels she is authorized to access.

Fig. 3.8
Using Groups to Give Access

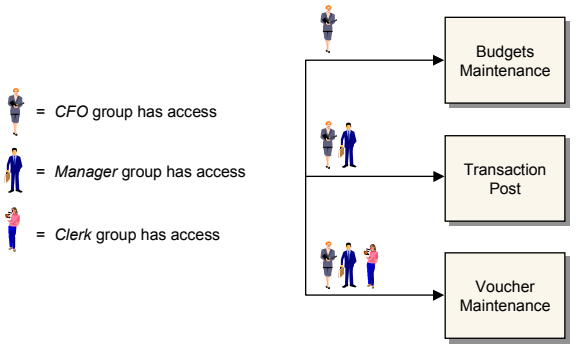


Table 3.3 shows how you would set up users in User Maintenance.

Table 3.3
Sample User Setup

User	Group
Helen	*Clerk*
Don	*Clerk* *Manager*
Sara	*Clerk* *Manager* CFO

Grouping users has two advantages. It simplifies access for the user and reduces maintenance for the system administrator.

When a new clerk is hired, the system administrator only has to assign the user to the *Clerk* group. The administrator does not have to use Menu Security Maintenance and add the new user as an acceptable session ID for all the protected AP functions.

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.

Using Field Security

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” on page 51.

Field Security Validation

In the standard release of MFG/PRO, security is not active for any fields, and only a few fields are eligible for field security. Use the Dictionary Field Security Report (36.3.6) 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. For eligible fields, the message Password Validation displays.

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 (36.3.5) to see which fields have security activated. It also lists privileged user IDs.

Dictionary Field Security Report

The Dictionary Field Security Report (36.3.6) 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 (36.3.4).
2. Verify that the field is secured by running the Activated Field Security Report (36.3.5).

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 (36.3.4).

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 (36.3.4).

9. Verify that each field is secured by running the Activated Field Security Report (36.3.5).

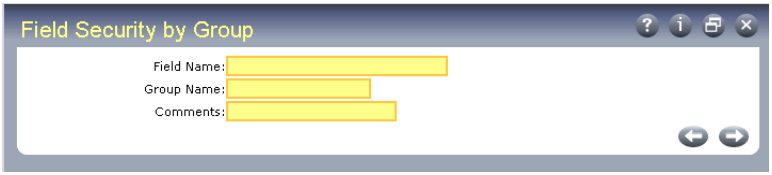
Note For multi-language installations, you must run `mgfldcmt.p` against your master database. Then, when you recompile, follow the normal procedure for compiling a multi-language version, including the creation of a translated database to compile against.

Field Security by Group

You can also set up field security for a group of users.

1. Assign users to groups in User Maintenance (36.3.18).
2. Execute Field Security by Group (36.3.7). This function adds all users who belong to a specified group to the list of authorized users for a validated field.

Fig. 3.9
Field Security by Group (36.3.7)



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 (36.3.4) 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.

Once Field Security by Group is executed for a field and group, all users who belong to the group display in Field Security Maintenance (36.3.4) 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.

Using Site Security

Site security lets administrators control user access to inventory transactions at each site. 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 (36.3.16).

Fig. 3.10 Site Security Maintenance (36.3.16)

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, 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 database individually.

Implementing Site Security

Because of the complexities of MFG/PRO 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 (36.3.16) cannot complete transactions at secured sites.

See “Specifying Groups” on page 51.

To implement site security, associate groups with users in User Maintenance. You can use previously defined groups to implement site security.

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.

Using Entity Security

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 all 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 3.11.

Note Entity security cannot be defined for groups.

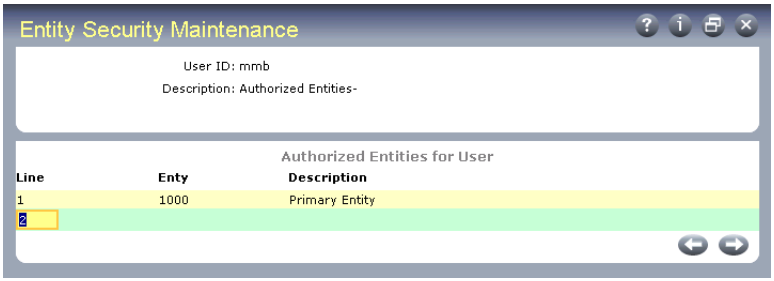


Fig. 3.11
Entity Security Maintenance (36.3.13)

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 (*).

Using GL Account Security

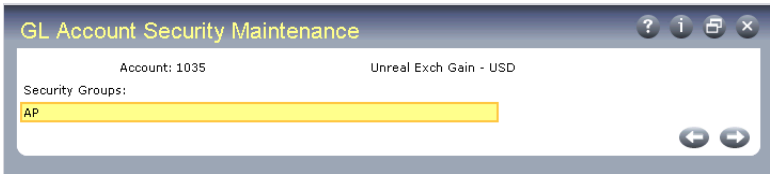
GL account security lets you restrict who can update GL accounts. Account security must be set up by user group. Account security is only effective when Verify GL Accounts is Yes in System/Account Control (36.1).

Account security groups are validated against groups assigned to users in User Maintenance (36.3.18).

See “Specifying Groups” on page 51.

Use GL Account Security Maintenance (36.3.9) to assign security groups to account numbers. Use the GL Account Security Report (36.3.10) to list all accounts that have security groups.

Fig. 3.12
GL Account Security (36.3.9)



When a user attempts to create a transaction affecting an account, the system checks the list of 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.

Account security is not applied during Transaction Post. Use Menu Security Maintenance (36.3.1) to restrict posting functions.

Using Inventory Movement Code Security

Use Inventory Movement Code Security (36.3.11) to grant or deny access to individuals and groups when using a specific inventory movement code at a particular site.

Fig. 3.13 Inventory Movement Code Security (36.3.11)

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 Volume 2A: 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 (36.3.12) 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.

Chapter 4

System Interface

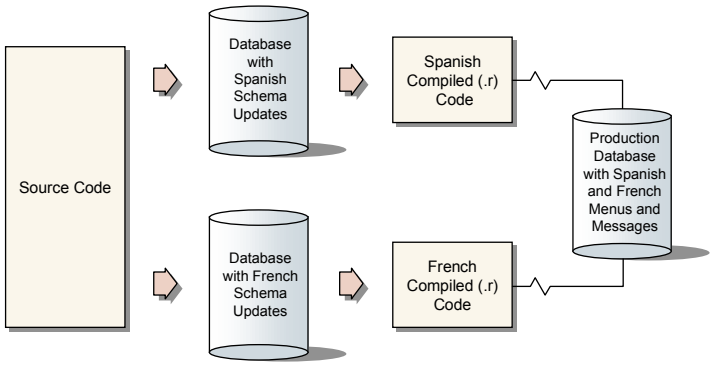
The System Interface menu contains programs that control menus, messages, multi-language installations, and help. If you are using QAD Desktop, interface details are discussed in the *User Guide: QAD Desktop*.

Using Multiple Languages

MFG/PRO 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 MFG/PRO in Spanish and others who want to see it 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).

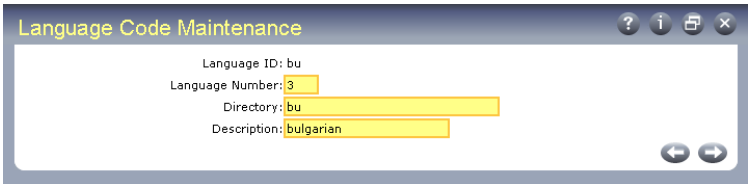


Fig. 4.2 Language Code Maintenance (36.4.1)

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 (36.3.18). This ensures that when the user logs on, MFG/PRO calls the Progress programs for that person's language.

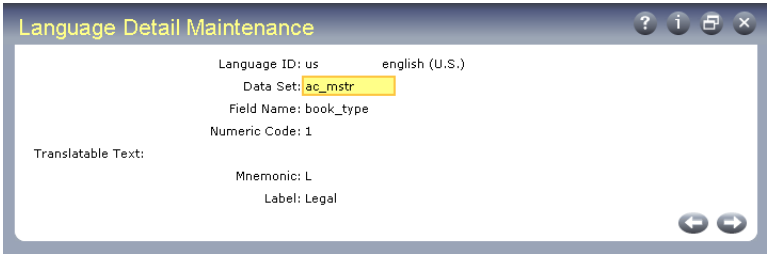
See "Defining Users" on page 42.

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 in MFG/PRO 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.

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

See “Using Menu Security” on page 50.

All menu information is contained in the mnd_det table. View its structure in the MFG/PRO 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.

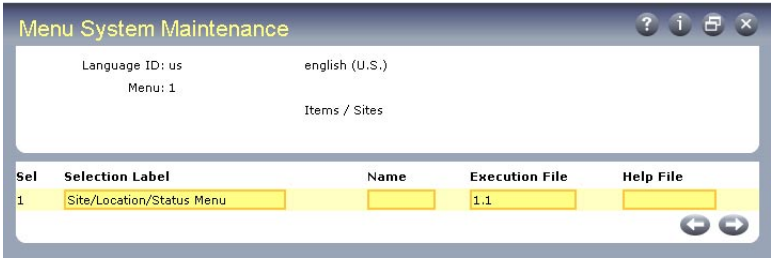


Fig. 4.4
Menu System Maintenance (36.4.4)

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.

See *Installation Guide: QAD Desktop*.

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.

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 184.

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 MFG/PRO, 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.

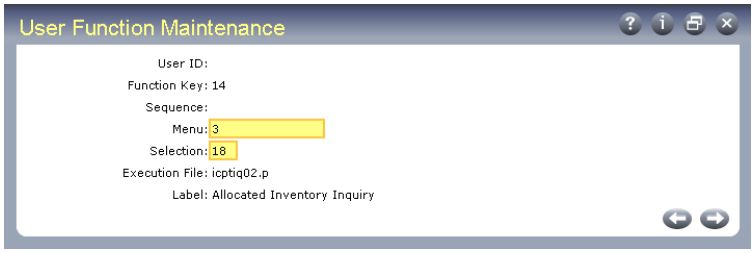


Fig. 4.5
User Function Maintenance (36.4.11)

Modifying Labels

MFG/PRO 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.

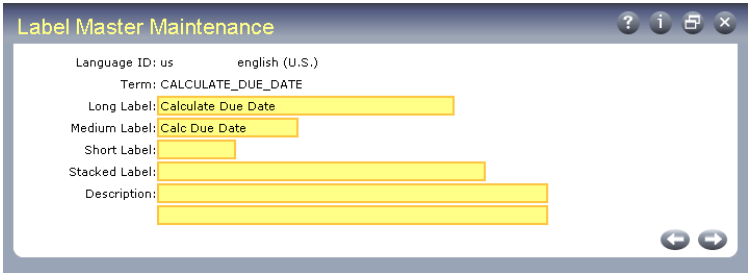


Fig. 4.6
Label Master Maintenance (36.4.17.1)

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

MFG/PRO has two kinds of error 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 MFG/PRO, and an MFG/PRO program error message is substituted, so this should occur rarely.

You can modify MFG/PRO 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.

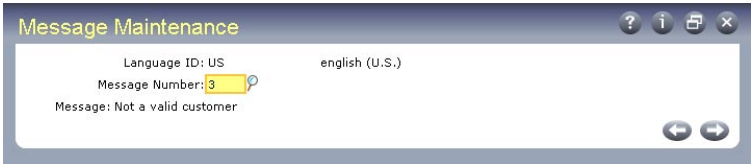


Fig. 4.7 Message Maintenance (36.4.7)

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 by MFG/PRO in a later version.

Using Field and Procedure Help

MFG/PRO 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 or character format. The content of the Windows and character help files is identical. However, you can add your own information to the character help files.

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

In the Windows interface, online help uses the standard Windows help format (`.hlp` files). In addition to receiving context-sensitive help on the active field and procedure, you can also access other help topics through the help menu, hypertext links, and keyword searches.

In the character interface, view field help by pressing F2 with the cursor in the field. Press F2 a second time and procedure help displays. No keyword searches or hypertext links are available in character help.

Adding User Help

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

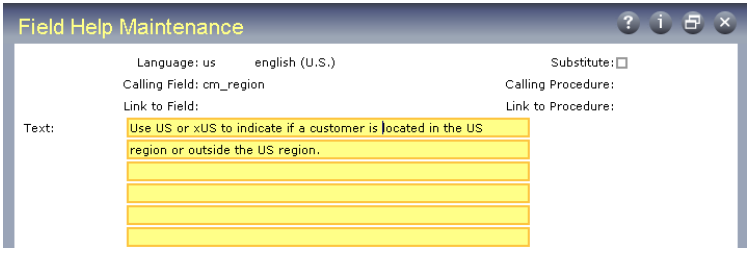


Fig. 4.8 Field Help Maintenance (36.4.13)

See “Setting Interface Preferences” on page 47.

Important Currently, you can only update help viewed in character sessions. If you are using a Windows environment, view character help by turning off the WinHelp setting in User Maintenance (36.3.18).

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 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 MFG/PRO 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 from MFG/PRO. Then, in User Maintenance (36.3.18), specify an e-mail definition and address for each user.

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.

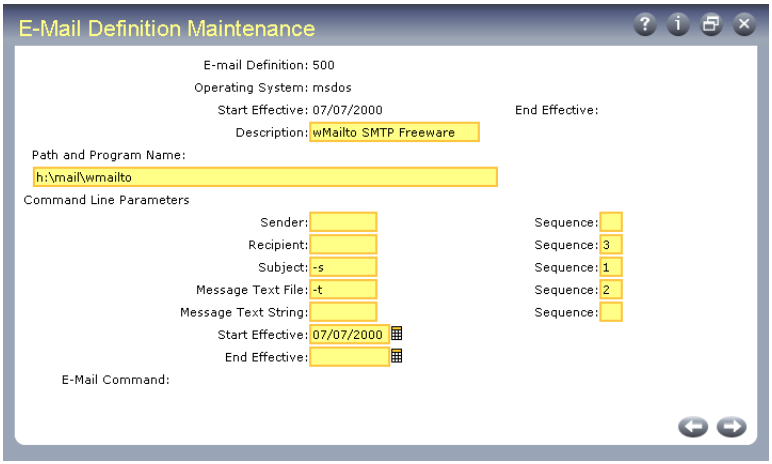


Fig. 4.9
E-Mail Definition Maintenance (36.4.20)

E-Mail Definition. 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 MFG/PRO 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 (36.3.18) for each user: E-Mail Address and Definition.

See page 42.

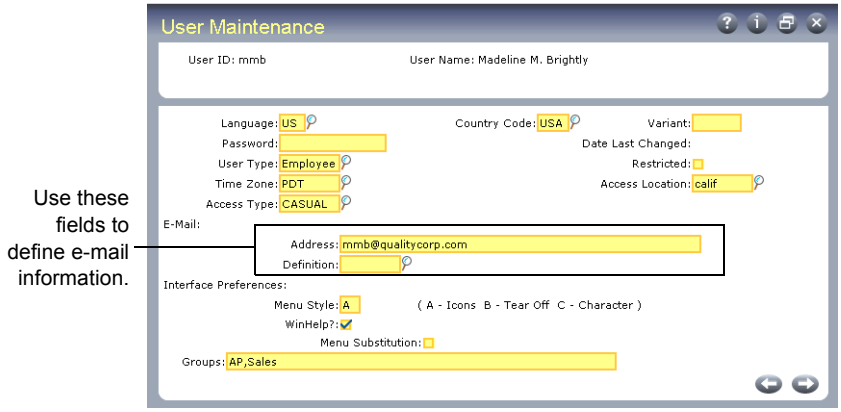


Fig. 4.10
User Maintenance (36.3.18)

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.

Chapter 5

Multiple Databases

This chapter covers the implementation and administration of multiple databases used with central order processing, distributed purchasing, or distributed requirements planning (DRP).

Introduction

This chapter describes how to implement consolidated order processing and DRP in a multiple database environment. Only issues relevant to multi-database implementation are addressed. For information on:

- Consolidated order processing, see *User Guide Volume 2A: Distribution*.
- Distribution requirements planning, see *User Guide Volume 5: Supply Chain Management*.

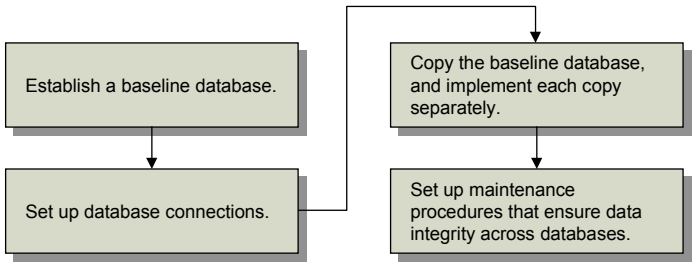
Each MFG/PRO database contains the following:

- A set of tables containing base data for customers, items, and sites
- A set of site-based inventory tables
- A security system
- A chart of accounts
- A primary entity

In multi-database operations, some of these elements must be the same across databases, others must be different across databases, and still others can be either the same or different. Moreover, in some databases some tables are not used, and precautions must be taken to prevent them from being used.

To accomplish this, first set up a baseline database containing information common to all databases you intend to connect. After establishing database connections, copy the baseline database as many times as necessary, and implement each copy separately. Finally, set up procedures for maintaining data integrity across databases.

Fig. 5.1
Multiple Database Setup



Establishing a Baseline Database

To establish a baseline database, do the following.

1. Make a copy of the *mfg* database that comes with MFG/PRO. Load your data into this database.
2. Use Language Code Maintenance (36.4.1) to set up the language version you plan to use.
3. Set up data common to all databases to be implemented, as shown in Figure 5.2.

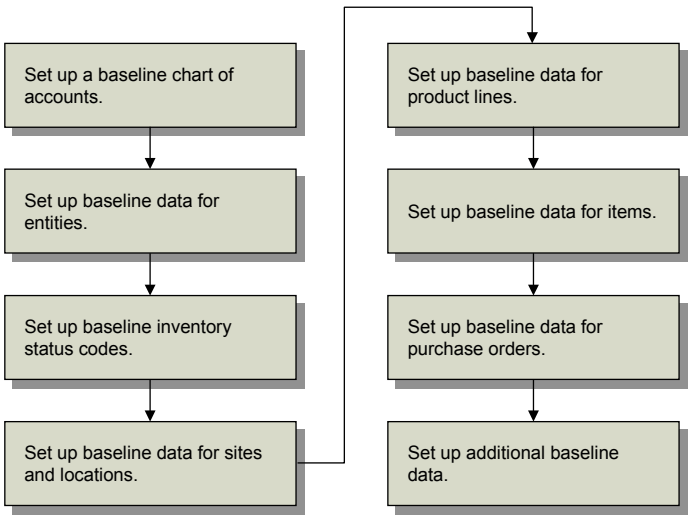


Fig. 5.2
Multiple Database Data Setup Work Flow

Setting Up a Baseline Chart of Accounts

As a general rule, the more accounts shared across databases, the easier the implementation. At minimum, make common any accounts used with:

- System/Account Control (36.1)
- Intercompany transactions
- Inventory
- Product lines
- Accounts Receivable

- Accounts Payable
- Purchase orders
- Sales orders

  Note Keep sales order setup information such as customers and price lists in the sales order database only.

Some countries, such as France, require a specific chart of accounts in financial statements, which may not match the operational database's chart of accounts. In this case, it may be useful to run two databases.

- Use an inventory database for operational, day-to-day control and intercompany reporting.
- Use a financial reporting database for general ledger (GL) data. This database contains account cross-reference data, which permits easy translation from the inventory database.

Set up the default accounts for Sales and Cost of Goods Sold in Sales Account Maintenance (1.2.17). These can be modified after you copy the baseline database.

Intercompany Transfer Accounts

Establish intercompany debit and credit accounts by entity in Entity Code Maintenance (25.3.1.1) for accounts payable, accounts receivable, inventory control, and fixed assets transactions. Each entity can have its own set of intercompany accounts. These accounts are used to track amounts for any transaction between entities. If you are using DRP, you can also set up accounts for transfer clearing, transfer variance, and goods in transit. Specify the default Transfer Clearing account in Inventory Control (3.24).

You can specify default intercompany accounts in the System/Account Control (36.1). Defining default accounts simplifies the setup of intercompany accounts in Entity Code Maintenance for multiple entities.

Tip
Create bank codes in Bank Maintenance (28.9.1) first.

Also, create a default bank for accounts payable. You may need to change the default bank later for individual databases.

Setting Up Baseline Data for Entities

An entity is a company or a way of grouping financial reports. Entity codes must be identical across databases so that appropriate financial transactions can be posted in each database. Define entities with Entity Code Maintenance (25.3.1.1).

You can record financial transactions in any currency. Transactions in different currencies are reconciled in the same way as in single-database operations.

Tip
If you use multiple currencies, set up currency codes first in Currency Maintenance (26.1).

Note While you can create transactions in multiple currencies, all entities within a database share one base currency defined in the System/ Account Control.

After copying the baseline database, you can change the primary entity and base currency of each copied database. They do not need to conform to those of the baseline database.

Setting Up Baseline Inventory Status Codes

Set up inventory status codes with Inventory Status Code Maintenance (1.1.1) so they can be used at any site in the system. After copying the baseline database, you can change or add codes for each copy.

Set up the following special codes.

- One that is fully restricted and does not allow use of any transaction codes. Use this as the default inventory status code at sites that do not belong to a database.
- One for the inter-site transportation location, if you are using DRP. Make the code non-nettable, since DRP orders are already counting the inter-site shipment as supply.

Setting Up Baseline Data for Sites and Locations

Site codes must be identical across databases. Set up sites and locations so that inventory tables in a database not belonging to a site cannot have inventory information associated with that site.

Each site must be assigned to just one database, and be unique across all databases. When the system looks for information on a site, it should find that information in a single database.

For each site, assign an address to the site code in Company Address Maintenance (2.12). The site address can be used as a ship-to address when items in a purchase order go to different sites and by freight lists when computing freight charges.

Set default sites and the costing methods in Inventory Control. You can change these settings later for individual databases.

Locations must be identical across databases. Individual locations can be added later and attached to sites. You can disable automatic locations for some or all sites by setting up default locations.

If you know your transportation mode, set up locations with names identical to the transportation mode in each DRP site.

Setting Up Baseline Data for Product Lines

Product lines to which purchased or finished products are assigned must have identical account codes across databases. Other product lines can be added after the baseline database is copied.

Setting Up Baseline Data for Items

Item codes and units of measure must be identical across databases. Enter into the baseline database all information pertaining to items sold, purchased, or transferred between sites using DRP. Items can be added to individual databases later.

An item must be associated with a site, but you can change this association after copying the baseline database. A default cost set for an item-site is located in the database in which it is created.

Associate centrally purchased items with the central purchase order site. Items purchased locally can be associated with a site at any time.

Setting Up Baseline Data for Purchase Orders

Purchase order setup information such as suppliers, supplier items, credit terms, tax codes, and currency must be identical across databases. Settings in Purchasing Control (5.24) should be identical except for the next purchase order number. Numbers must be unique across databases, so you should identify a unique prefix or separate range for each database that creates orders.

Setting Up Other Baseline Data

Make significant base data such as unit of measure conversions, site calendars, currency and conversions, printer types and printers, banks, product structures, and format positions identical across databases.

DRP setup information including source networks, transportation modes, and transportation schedules must be identical across databases. DRP networks and sourcing relationships can be defined later, but it is important to make these consistent among sites. Set up transportation modes and shipping schedules if known and if consistent among sites.

Setting Up Database Connections

Figure 5.3 illustrates the steps to connecting databases.

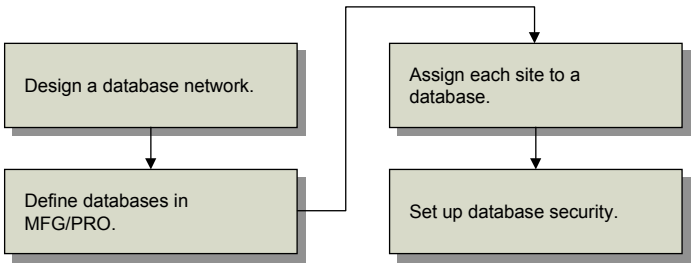


Fig. 5.3 Database Connection Flow

Designing a Network

For instructions on setting up networks, see *Progress System Administration Reference*. Any network supported by Progress is acceptable, but all databases must be on the same type of network.

See "Managing Multiple Databases" on page 92.

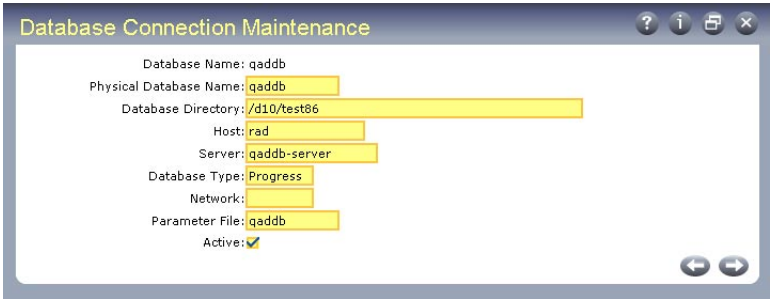
Choose a host name, server name, and path name for each database. This information can be changed later.

Defining Databases in MFG/PRO

Use Database Connection Maintenance (36.6.1) to specify which databases are to be connected.

Note The preferred method for maintaining database connections is using features of MFG/UTIL. These are described in the appropriate installation guide for your system.

Fig. 5.4 Database Connection Maintenance (36.6.1)



Database Name. Enter the name of an MFG/PRO database. Database Name displays at the top of each menu.

This name is usually the same as the physical database name, but does not have to be. If it is not, the `-ld` parameter should be used in the startup script. Enter the value specified here following the `-ld` parameter.

For the primary database, you do not need to specify a physical database, host, or server. However, you must enter a value for Database Name. You must specify this name in the Database field for all sites in the current database in Site Maintenance (1.1.13).

Physical Database Name. Enter the Progress name of the database, such as `qaddb`. This database name is case-sensitive, since it is passed to UNIX. There is no extension.

Database Directory. Enter the full path name of the directory that contains the database specified in Physical Database Name. The name should not be followed by a slash (/). Only fields used to locate the database are required. If all databases are in the same directory, for instance, you do not need to supply the directory name.

Tip
The path name is case-sensitive.

Host Name. Enter the name of the host exactly as it appears in the `/etc/hosts` (UNIX) file (or equivalent) on the machine where the current database resides. If the host parameter is specified, specify a server parameter also. If the host is the current host, neither host name nor server need be specified, except with TLI networks on UNIX V.4.

Server. Enter the name of the server or broker used when starting up the remote databases. Must be identical to the server name specified in the `/etc/services` file. This value is not needed if the database is on the current machine.

Database Type. The default value is Progress.

Network. Enter the type of network you use. Only TCP, DECNET, NETBIOS, SPX, and TLI are supported.

Parameter File. Enter the exact name of the optional parameter (`.pf`) file to be used by the primary database to find the named database. Parameters specified in parameter files supersede settings made in this program. The `.pf` suffix is neither required nor supplied. The parameter file must be in the PROPATH.

For details on using `.pf` files, see "Managing Multiple Databases" on page 92.

Database Connection Maintenance and parameter files overlap. The decision whether or not to use parameters files depends on how you choose to handle system security. The format of parameter files and the meaning of connection parameters are discussed in the Progress documentation.

See "Startup and Shutdown" in the *Progress System Administration Guide*.

Active. Enter Yes to have the primary database attempt connection with this database at startup. Otherwise, enter No.

Assign Each Site to a Database

During transaction processing, the system determines which database to use based on the value associated with the site in Site Maintenance (1.1.13). Your input in Site Maintenance is validated against databases defined in Database Connection Maintenance.

After assigning databases, you can normally complete inventory transactions only for sites associated with the current database.

Set Up Database Security

See “Managing Multiple Databases” on page 92 for details.

The system checks the security records in each database when validating user access to menus, fields, GL accounts, entities, sites, and inventory movement codes. When users access data in multiple databases, they typically need access to the same programs. In this case, set up security in the primary database. You can then modify security for individual databases after you copy them, if needed.

Copying and Connecting Databases

See the installation guide for more information.

1 Copy databases to the appropriate locations on your network using `prodb` or MFG/UTIL. Use physical database names and directories, as defined in Database Connection Maintenance (36.6.1).

2 Start the servers for each database using the following command.

```
prostart <dbname>
```

See “Login and Connection Methods” on page 92 for a sample script.

3 Check connections by logging into a database using a login script. Use Database Connection Inquiry (36.6.2) to review connections. You can work in any database during a single session by using Current Database Change (36.6.17) as needed.

Setting Up Other Databases

Once your database network is running, you can make changes to individual databases as needed.

Maintaining System Accounting Data

Change the system entity and base currency as needed in System/Account Control (36.1). If needed, use Entity Code Maintenance (25.3.1.1) to change the primary entity to the system entity.

If your inventory database uses a different base currency from the central database, use Account Code Maintenance (25.3.13) to set up Sales and Sale Discount accounts with a foreign exchange conversion index of 4 (historical).

Maintaining Locations

Normally, the sites associated with a particular database should not be accessed in other connected databases. You can use the automatic location feature to ensure that new locations are not created inadvertently. To do this, set Automatic Locations to No in Site Maintenance (1.1.13) for sites not belonging to the current database.

In most inventory transactions, inventory is placed in a site/location pair. When Automatic Locations is No, new pairs cannot be created during transaction processing. Since no pairs currently exist in a new database, this prevents inventory from being moved to or from a site accidentally.

Maintaining Inventory Status Codes

For sites not associated with the current database, assign a fully restricted inventory status code as the default.

Maintaining Sales Orders

Set up Sales Orders/Invoices in the sales order database. Enter base data such as customers, salespersons, and freight lists.

See *User Guide Volume 2A: Distribution* for details.

Maintaining Items and Sites

Use Item Inventory Data Maintenance (1.4.5) to assign sites to inventory items in each local database.

Maintaining Item Planning and Cost Data

Set up planning and cost parameters for items at each site in a database. Assign DRP items a purchase/manufacture code of D, and specify a source network.

Maintaining Purchase Order Sites

Assign a purchase order site to each purchased item. This is the site that issues purchase orders for this item.

Maintaining Prefixes

In Purchasing Control (5.24) and Sales Order Control (7.1.24) in each database, set unique document prefixes for the database. This quickly identifies sales orders created in the wrong databases.

Maintaining Security

Set up security for each database. You may want to restrict access to Current Database Change (36.6.17) and Database Connect (36.6.13).

In Security Control (36.3.24), create a unique session ID prefix for each database. This prevents conflicts when temporary files are written, and improves security by uniquely identifying users.

Maintaining Printers

Modify the printers for each database as needed. In sales and purchase order databases, define printers for shipment information. If the printers are already connected to the sales order database's CPU, set up printers as usual. If the printers are on another CPU in the system, define the printers in the sales order database, and refer to your network guide for information on printing to another CPU.

To print Database1 sales orders through the Database2 CPU, use Printer Setup Maintenance (36.13.2) to give the Database2 printer a name and description. In the Device Pathname field, specify the operating system commands used to send printer output to the Database2 printer.

Example Many UNIX systems on an Ethernet network use `remsh` to execute a command on a remote system. If Database2 is located on a CPU called db2, enter the following command for Device Pathname:

```
remsh db2 'lp -drelp -s'
```

To simplify printing, use Printer Default Maintenance (36.13.4) to set printer defaults by user ID. To do this, user IDs must be unique across databases. Do not set default printers by menu item alone, if different picklists may need to be printed at different sites.

Modifying Startup Connections

Use Database Connection Maintenance (36.6.1) to set default connections for each database. If you do not want databases connected automatically at startup, set Active to No.

Modifying DRP and MRP

Add network codes, if necessary. Check transportation networks and shipping schedules. Create locations with the same name as the transportation modes for each site, making status codes non-nettable. Check settings in the DRP and MRP control programs.

For each database involved in intersite requests, create subdirectories under the database directory using the names of the databases you connect to. Give them write permission. Intersite request records are written to these subdirectories when a connection is broken.

Managing Multiple Databases

Multi-database administration involves ensuring that connections are maintained between databases and that users have proper access to data and functions. The administrator determines the following:

- Database locations on CPUs. Progress must be installed on each CPU, and each must be connected by a network supported by Progress.
- Server locations and how servers are started. A server must be running for each database.
- Which MFG/PRO modules are activated in each database.
- User access to databases. Do this by assigning user IDs and login IDs for database CPUs, specifying a primary database for each user, and controlling user activities in the database.

Login and Connection Methods

Initial logins to the primary database and connections to other databases can be managed in two ways.

1. Log in to the primary database with a shell script, then use Current Database Change (36.6.17) to switch databases.
2. Log in to each database using a different login script.

On continuously connected networks with servers for all databases running, you can log in to any database.

Example To log in to Database2 located on the Database2 CPU, a log-in script includes the following.

```
exec $DLC/_progress /qad/database2 -H database2
-S database2 -p mf.p -ld database2
```

Table 5.1 shows the login script parameters.

Table 5.1 Sample Login Script Parameters

Parameter	Description
exec	UNIX command that runs Progress.
$DLC_progress	Version of Progress on your system.
/qad/database2	Full path name of Database2.
-H database2	Host containing `/qad/database2` (in `/etc/hosts` file).
-S database2	Server connected to (in `/etc/services` file).
-p mf.p	Progress program that launches MFG/PRO.
-ld database2	Logical name of the database. If a `-ld` parameter is specified, the value given for that parameter must also appear in Database Name in the entry for the current database in Database Connection Maintenance. Also, sites attached to the primary database must include the logical name in the Database field.

In the script, the value of the `-ld` parameter (Database2) sets the logical name of the primary database. To determine if a site is in the current database, the system checks the database name associated with the site and compares it with the value of the `-ld` parameter. If they are the same, the site is in the current database.

For this reason, each database name used in an `-ld` parameter must be defined in Database Connection Maintenance. Since the login script does the actual work of connection, you do not need to enter values for Physical Database name.

When the primary database starts, the system tries to connect to every other database by issuing connect statements for each database with Yes in the Active field of Database Connection Maintenance.

You can establish the connections directly in the Progress log-in script, as shown in the following example.

```
exec $DLC_progress /qad/database2 -H database2
-S database2 -ld database2 /qad/database1
-H database1 -S database1 -ld database1 -p mf.p
```

The command tells Progress to run `mf.p` against two databases, each of which is located on a different computer. Since Database2 is mentioned first, it is the primary database. The first `-ld` parameter provides the logical name of the primary database.

You can also specify all connection parameters such as `-H` or `-S` in a parameter file. The following command has the same effect as the previous login script.

```
exec $DLC_progress /qad/database2 -pf database2.pf /qad/database1
-pf database1.pf -p mf.p
```

If you use Database Connection Maintenance (36.6.1) to connect to Database1, you can specify `database1.pf` in the Parameter File field. The parameter file substitutes for and overrides individual specifications for each parameter.

Both methods achieve the same result. The user is logged into Database2, which is the primary database for that user. Actions taken in the primary database affect data in that database only, except where MFG/PRO accesses other databases to retrieve or record information.

To alter data in another database directly, use Current Database Change (36.6.17). Data in that database is affected by user activity. Data in other databases is affected only by consolidated order processing or DRP actions. When users change databases, the security records defined in the current database are checked to determine access.

Database connections can also be managed using Database Connect (36.6.13) and Database Disconnect (36.6.15). Databases must exist in Database Connection Maintenance, but can have Active set to No. These programs are most useful for resource-intensive operations like DRP. To keep DRP from overloading a connection between sites, you can disconnect the sites before running DRP. DRP then writes all records to a subdirectory, where you can access them later.

Controlling User Logins

Combinations of Progress and MFG/PRO connection methods let you control how users log in.

- In each database, use Database Connection Maintenance (36.6.1) to list all other databases. Control the use of databases with login scripts. Permit users who need to change databases to change the current database with Current Database Change.

- Leave Database Connection Maintenance empty or inactive, and control connections with log-in scripts. Since connections are associated with individual sessions, some users can connect to one database only, while others can connect to more than one.
- Leave the databases in Database Connection Maintenance inactive, and allow specified users to connect to other databases using Connect Database (36.6.13), as needed.

Example An administrator in a plant where Database1 is located creates two log-in scripts, one making Database1 the primary database, the other making Database2 the primary database. Table 5.2 shows the kinds of access personnel are given.

Table 5.2
Access Examples

Personnel	Access
Salespeople	Database1 log-in script. All sales orders filled in Database1.
Inventory Controllers	Database2 log-in script. Their work does not require access to other databases.
Master Schedulers	Database1 and/or Database2 log-in script, depending on security. Ability to access Current Database Change.
Shippers	Database1 log-in script. Shipments are recorded in Database1.
Purchasers	Either Database1 or Database2 log-in script, depending on site used.
Receivers	Database2 log-in script. Receipts are recorded in Database2.

In this example, users can log onto other computers from their computers, then execute log-in scripts. The log-in script for Database1 is identical in Database1 and Database2, except where the log-in ID for a user is different on different computers.

System Security

The ID and groups associated with a user are defined based on the user profile in the initial log-in database. These values are passed to each subsequent database during log-in. However, the system reads the security records in each new database whenever the user attempts to execute a secured function in that database. If you do not have the same security setup in each database, some users may not be able to execute the same functions in both.

See Chapter 3, “Users and Security,” for a full discussion of security issues.

Example User RBT belongs to group AP, which has access to all functions on the Accounts Payable (28) menu in Database1. In Database2, group AP can only execute AP reports. User RBT logs into Database1 and can update vouchers. After connecting to Database2, RBT can only execute voucher reports.

See "Registering Licenses" on page 136.

In addition to security, the system also checks licensing in each database. Users must have the proper license access to execute programs.

Connection Loss

Multiple database operation assumes databases are connected continuously. When database connections fail, consolidated order processing capabilities may be lost. In this case, users are notified.

You cannot enter a sales order referencing a site in a disconnected database. This is true of both unconfirmed and confirmed orders.

Since entering a purchase order typically causes a requisition in another database to be erased, this action cannot be performed when connections are down. For most purchasing transactions, both databases must be updated at the same time. If the system fails during a transaction, corrections must be made manually in both databases.

Multiple Database Example

When implementing multiple databases, you must ensure that information is available to databases where and when it is needed, and that the information is consistent across databases. Assume, for instance, you create a sales order in a sales order database, with a line item to be shipped from an inventory database.

The sales order uses master data such as customers, salespersons, items, credit terms, freight lists, and price lists from the sales order database.

When the site belonging to the inventory database is specified, the system looks at the site master, determines which database the site belongs to, then checks the database connection master (dc_mstr) to see if that database is connected. If it is, the system processes this and all subsequent

transactions involving the site by accessing the inventory database. When the order is created, allocations and demand records for the site are created in the inventory database.

When the item is shipped, inventory records, item master, and demand records in the inventory database are updated. The inventory transaction occurs in the inventory database, and the associated GL records are created in that database, using accounts specified by that item's product line. Item costs are obtained from item-site cost records in the inventory database.

When the invoice is posted, tables such as customer master and credit terms in the sales order database are used, and accounts associated with those tables are used for GL transactions. These GL transactions appear in the sales database.

Purchase orders follow a similar sequence, again driven by the fact that each site is assigned to a single database.

Chapter 6

Printers and Batch Processing

This chapter describes how to set up and use printers in MFG/PRO.

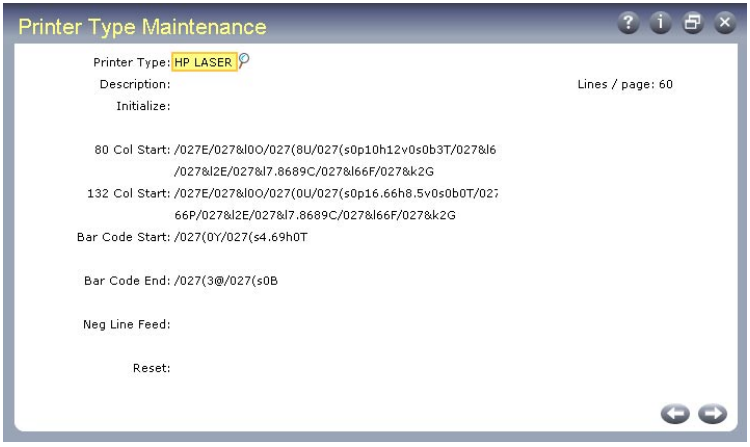
Introduction

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. The Batch Processing menu includes programs for creating batch print requests.

Defining Printer Types

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

Fig. 6.1
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.

Tip
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 6.1 lists characters frequently used in control sequences.

Table 6.1
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.

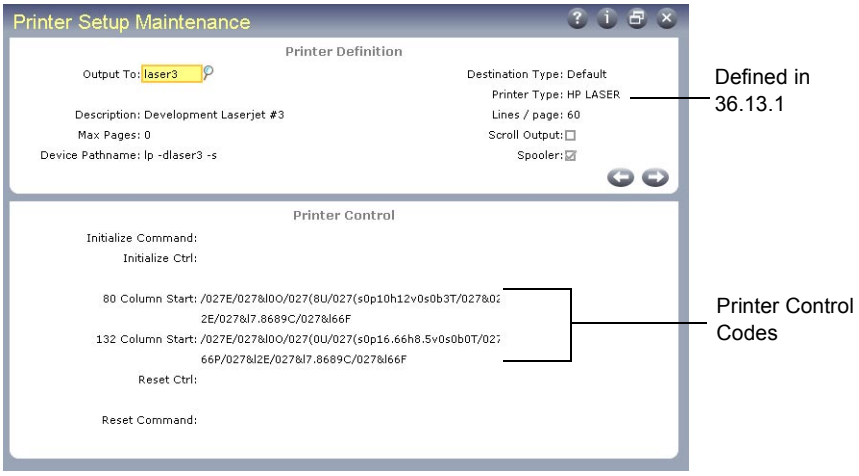
Table 6.2
Sample Printer Control Codes

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. 6.2
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 MFG/PRO 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

See “Building an E-Mail System Interface” on page 74.

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.

- 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.

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 6.3 lists examples of device path names.

Table 6.3 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.

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

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

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.

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

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 QAD Desktop

If users generate reports from the QAD Desktop 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.

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

Tip
Remember this is only the default; you can change it to any valid device when you run the program.

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 (GUI 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 MFG/PRO program to define alternate document formats. Instead, you must create a Progress program to generate them. Use the following steps to do this.

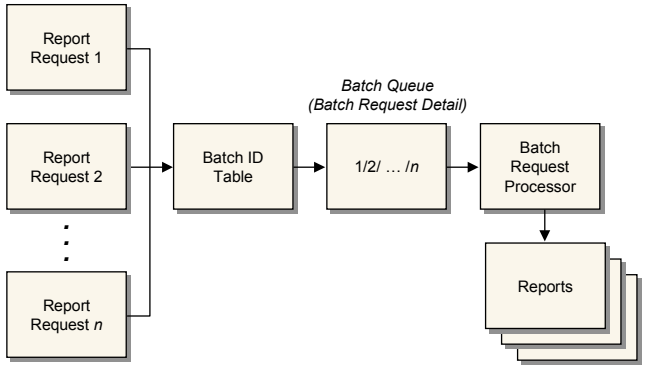
1. Create a Progress program to format the document as required.
2. Name the new program file appropriately so it can be located by the print program. The file name is typically created by removing the first two characters of the print program name and appending a two-character form code.
3. Modify the 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. To set up a batch process, first create a batch ID in Batch ID Maintenance (36.14.1). Then select reports or programs that can be run in batch mode and submit those programs using the batch ID. You batch a report or process by specifying a batch ID for output rather than a printer ID.

Fig. 6.3
Batch Processes



Use Batch Request Processor (36.14.13) to run reports and/or programs submitted using a batch ID. You can process up to 10 batch IDs in a single run.

Tip
Use names that are descriptive and easy to remember, such as Paycheck, Monthly, or After5.

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.

Use Batch Request Detail Maintenance (36.14.3) to view reports and programs submitted to any batch.

You can set up batch files that run the batch processor, and from UNIX you can execute these jobs automatically. To set up a batch file, use the Progress commands `bpro` or `mpro`. The `mpro` command has the following structure:

```
mpro DB name -p <Progress prog name> startup parameters
```

The `<Progress prog name>` designates a file that includes commands such as the following:

```
Input From <input file name>
Output To <output file name or /dev/null>
run mf.p
Input close
Output close
```

The `<input file>` should use the CIM format, anticipating all data entry including login and logout:

```
"password"
mgbatch.p
"batchid"
.
.
.
.
"Y"
```

The four dots are exits. Y confirms the exit from MFG/PRO. You can run the batch file (the `mpro` program) automatically with `cron`.

Chapter 7

CIM Interface

This chapter describes how to use MFG/PRO 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 MFG/PRO data. There are three basic ways to transfer data into and out of your MFG/PRO database:

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

See page 123.

The first two options are discussed in Chapter 8. This chapter discusses CIM data load, which lets you load data into MFG/PRO from any source, as long as the data is formatted to match the MFG/PRO schema.

See “Deleting Records through CIM” on page 118.

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.

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.

See *External Interface Guide: Q/LinQ.*

Note Q/LinQ offers more advanced features for data transfer, including methods similar to CIM.

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.

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 MFG/PRO 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.

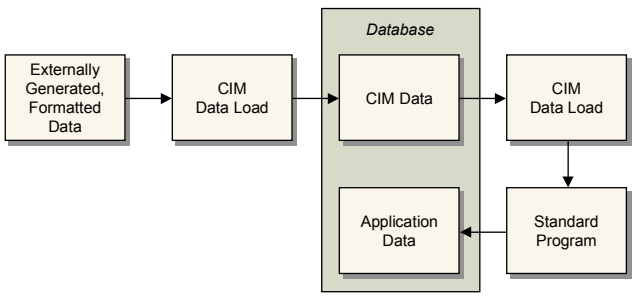
See “CIM Data Format” on page 112.

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.

Fig. 7.1
CIM Data Load

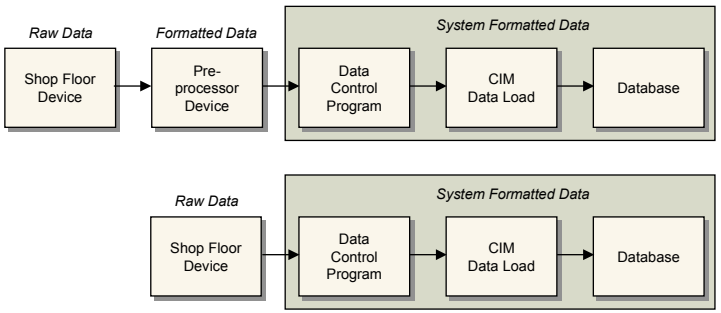


When CIM Data Load reads a data load group, it creates a record in the batch data load master table and assigns it a unique group ID. This integer record contains the name of the MFG/PRO program to receive the data, and the date and time when the record was added. CIM Data Load then creates a record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

See “CIM Data Format” on page 112 for details.

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.

Fig. 7.2
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

See "Determining Data for the Input File" on page 114.

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 "CIM Data Input File Example" on page 115.

The `@@BATCHLOAD` key word signals the beginning of the data-load group, consisting of one or more lines. Program name is the MFG/PRO 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.

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.
- 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. 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 01/01/02 for fields 4, 6, and 8, enter the following:

  ```
  --- "yes" - "12" - "01/01/02"
  ```

- 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.

Tip
The tilde (~) is not required if you create the CIM file in an editor.

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:

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

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

   a In the character interface, the name of the program displays in the upper left corner of the screen.

   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.

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, 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).

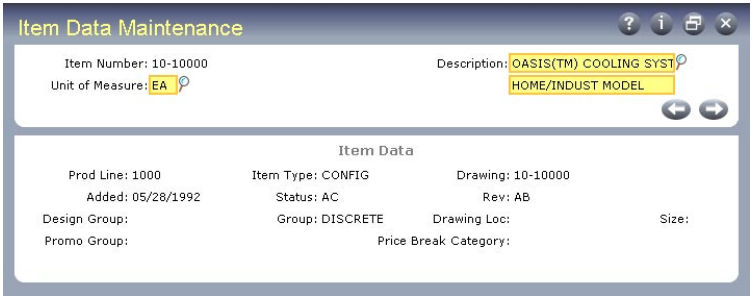


Fig. 7.3
CIM File for Item Data Maintenance (1.4.3)

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 MFG/PRO programs listed in Table 7.1. 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 message indicates that you can use batch delete. You can do this in the character and Windows interfaces.

Table 7.1 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

Menu Label	Program Name
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

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 MFG/PRO 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 7.1.

   Note In the character and Windows interfaces, press Ctrl+F to display the informational message: Batch delete is enabled. This message does not display in Desktop sessions.

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, MFG/PRO 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 8

Database Management

MFG/PRO 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

MFG/PRO provides utilities for managing the size of your database.

Determining Disk Usage

Tip
The program requires adequate free disk space to run.

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.

Use Disk Space Inquiry (36.24.13) 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. 8.1
Disk Space Inquiry (36.24.13)

```
/                  (/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.
- 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.

See “Deleting and Archiving Data” on page 127.

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 110 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.

For information on Progress dump/load and bulk load programs, see the Progress user manuals.

Note Progress and Oracle each provide dump/load and import/export programs, but these programs do not maintain the integrity of data in the MFG/PRO database.

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 124.

3. Log in to MFG/PRO 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 MFG/PRO 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 8.1 lists data that can be deleted and archived.

Table 8.1 Transactions that Can Be Deleted/ Archived

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

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 MFG/PRO 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.

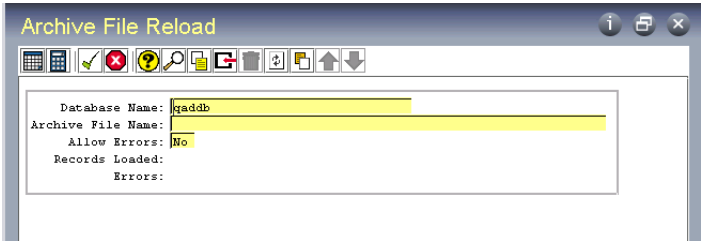


Fig. 8.2 Archive File Reload (36.16.5)

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 (36.3.18) to temporarily change your country code to match that of the user who archived the data.

See “Defining Users” on page 42.

Managing Database Sequences

When a unique identifier is needed by an MFG/PRO 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.

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

Tip
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.

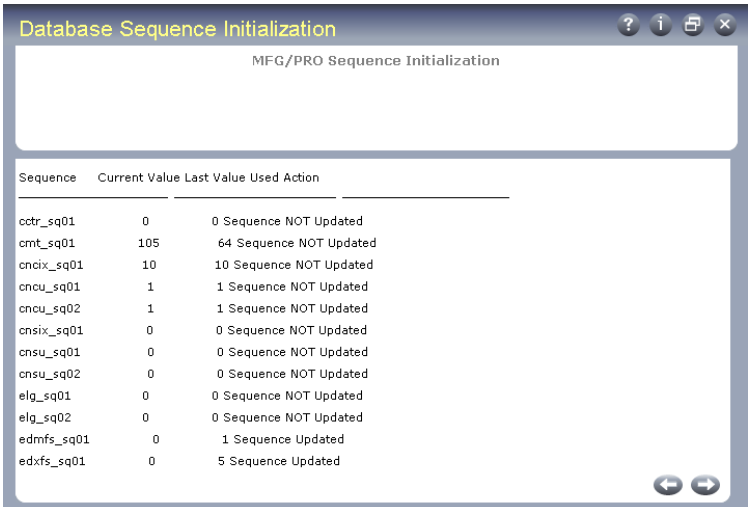


Fig. 8.3
Database Sequence Initialization (36.16.17)

Maintaining Sequences Manually

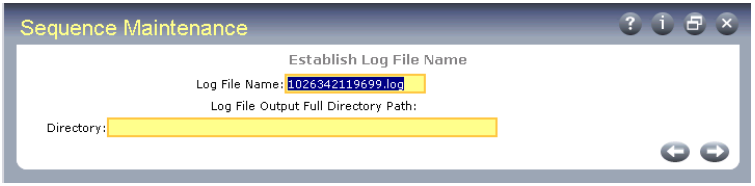
Maintain sequences manually or through the CIM interface. Maintenance includes:

- 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 135.

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. 8.4
Sequence Maintenance (36.16.13), Establish Log File Name Frame

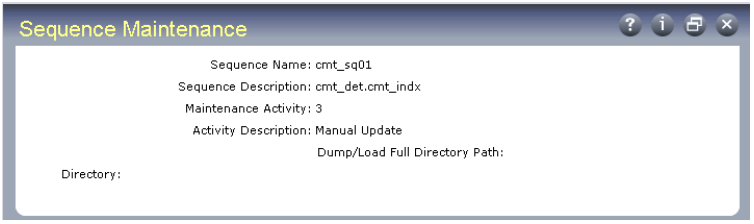


Log File Name. The name of the error log file.

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

A second Sequence Maintenance screen displays.

Fig. 8.5
Sequence Maintenance, Sequence Name Frame



Tip
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.

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

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.

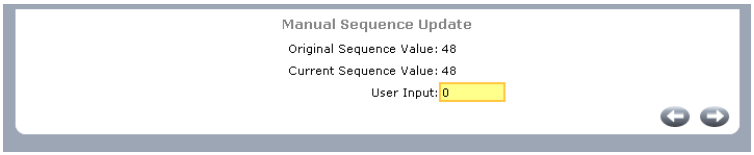


Fig. 8.6 Manual Sequence Update Frame

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

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 MFG/PRO and is part of the schema. An error displays when the value entered is not within the valid range.

Tip
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.

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 an MFG/PRO function. This value is not available for processing, since it was consumed by another process.

For more information on CIM, see "Using the CIM Interface" on page 110.

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

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 illustration:

```
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 CIM's MFG/PRO 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.

Tip
The default activity is dumping (1).

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

<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 MFG/PRO can be maintained using CIM.

Maintaining Audit Trails

The system maintains 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.

See “Maintaining Sequences Manually” on page 131.

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 an MFG/PRO function.

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 MFG/PRO software, you also receive license codes. This includes license codes for the MFG/PRO foundation 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 MFG/PRO, you must register the license codes with MFG/PRO.

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 MFG/PRO software to its customers for use by a predetermined number of users, sessions, or locations.

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

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

License Types

There are three license types in MFG/PRO:

Named User. Each unique user ID defined in User Maintenance (36.3.18) 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 139.

Location. Each user's access location is counted as a location. QAD licenses a predefined number of locations for specific applications. Customers must define their locations in Generalized Codes Maintenance (36.2.13). System administrators assign users an access location in User Maintenance (36.3.18).

See "Access Location" on page 44.

Table 8.2 is an example of the location license scheme. In the example, Logistics Accounting has a location license with three locations predefined.

Table 8.2 Location Licensing Schemes

Physical Users	Location User License Count
John accesses Logistics Accounting from the Seattle accounting center.	Counts as one location.
Mary accesses Logistics Accounting from the Los Angeles office.	Counts as two locations.
Angela accesses Logistics Accounting from the Pasadena distribution center.	Counts as three locations.
Bill accesses Logistics Accounting from the Los Angeles office.	Still counts as three locations.

User Counts

MFG/PRO software monitors license use regardless of your user interface type (character, Windows, or Desktop), 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 use Current Database Change (36.6.17) to switch to a new database, this process is repeated. This is because changing databases is like exiting your current database and starting a new MFG/PRO session. Whenever you switch databases, the system stores the logout date and time.

See *User Guide: QAD Desktop*.

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

See “Violation Messages” on page 140.

For named user license types, the software counts users when system administrators create new users in User Maintenance (36.3.18) 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.

License Violations

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

See“License Reporting” on page 145.

The system stores all license violation occurrences in MFG/PRO. 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, sessions, or access to certain programs from

additional locations. With warnings, messages display, but additional users, sessions, or locations can exist and users can still log in to MFG/PRO or separately sold MFG/PRO modules.

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.

See "Setting Up Security Control" on page 38.

Important The first time a warning displays, you can access MFG/PRO 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 to MFG/PRO if the license registration record does not exist for MFG/PRO. System administrators register MFG/PRO's license code in License Registration (36.16.10.1).

Violation Types

The system records the violation types listed in Table 8.3.

Table 8.3 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, sessions, or locations 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 with MFG/PRO

Violation Messages

Table 8.4 lists error messages that display when license violations occur.

Table 8.4
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 deactive some users; otherwise, contact your QAD representative or distributor to upgrade your license agreement.
Licensed location limit exceeded	This message displays in User Maintenance and License Registration when the number of locations from which users access the application exceeds the number of locations specified by your license agreement. Review the number and definition of the licensed locations per your contract and ensure that users have the correct location value assigned. To support more user locations, 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.

Message	Explanation and Solution
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.
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 MFG/PRO log-in, you cannot log in unless another currently logged-in user logs out.

Interaction with MFG/PRO

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

System administrators maintain defined named users, a list of registered software applications that users are authorized to access, and the access locations in User Maintenance (36.3.18). License registration software uses this information to prevent more active users or locations than the license allows.

See “Defining Users” on page 42.

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 65.

Multiple Time Zones Maintenance (36.16.22.1) lets you define valid time zone codes. The server time zone is defined in System/Account Control (36.1). System administrators associate users with a time zone in User Maintenance. The time zone information is useful when detecting problems with users in location license types.

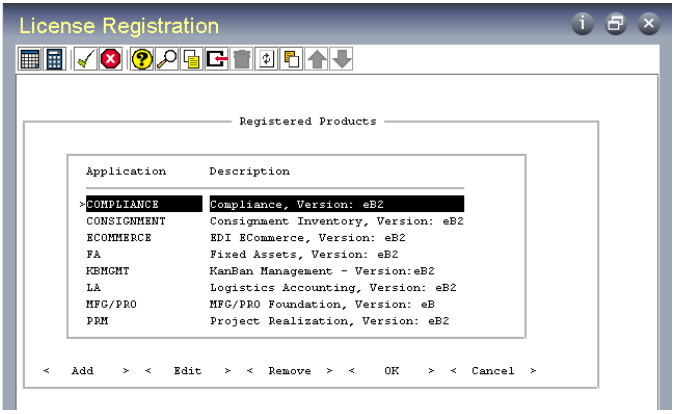
See “Setting Up Multiple Time Zones” on page 149.

License Registration

Use License Registration (36.16.10.1) to:

- Add a new license code for MFG/PRO software or separately licensed MFG/PRO modules.
- Upgrade license codes to add sessions, users, or locations.
- Remove license codes.

Fig. 8.7 Licensed Registration (36.16.10.1)



The system requests licensing information after you install MFG/PRO 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 MFG/PRO or an MFG/PRO module; then choose OK.

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

See “Granting Users Access to Registered Software” on page 143.

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 MFG/PRO displays once you enter a user ID.

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, sessions, or locations.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 MFG/PRO 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 (36.3.18).

See “Specifying Application Use” on page 49.

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. 8.8 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.

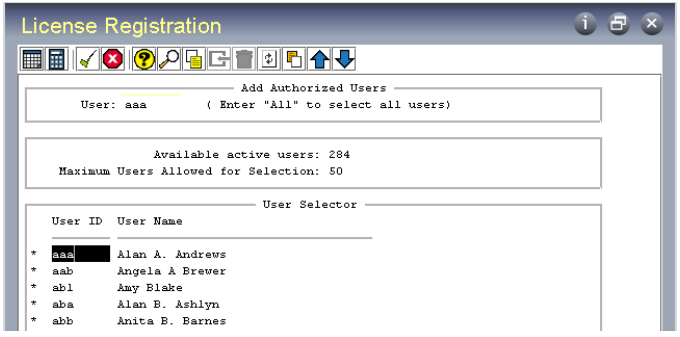
To de-activate users, press Spacebar. The asterisk is removed.

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 the total number of locations from which users access the application exceeds the number allowed by the applications license, a message displays and a violation is recorded.

If you need to authorize more users than your license allows, system administrators can add users through User Maintenance (36.3.18); however, the software records a violation of your license when you add more users.

Fig. 8.9 License Registration, User Selector Frame



License Reporting

MFG/PRO provides reports that 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), access location, and access activation date.

Licensed Application Report

Use Licensed Application Report (36.16.10.3) to display a list of software applications registered through the MFG/PRO 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.

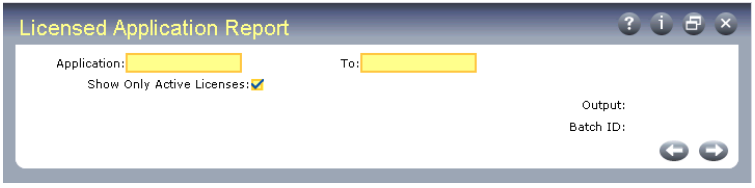


Fig. 8.10 Licensed Application Report (36.16.10.8)

Application Usage Profile Report

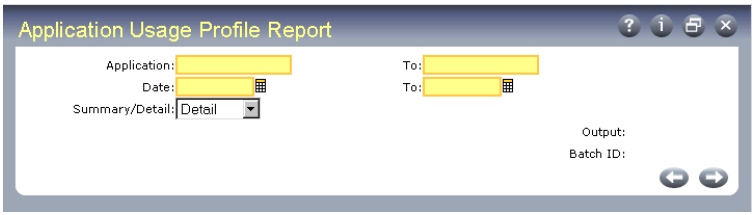
After you install and register an application with MFG/PRO, the software keeps statistics about your application use. The statistics include:

- Licensed application name
- Menu item's 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. 8.11
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)
- 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

See “Violation Types” on page 139.

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

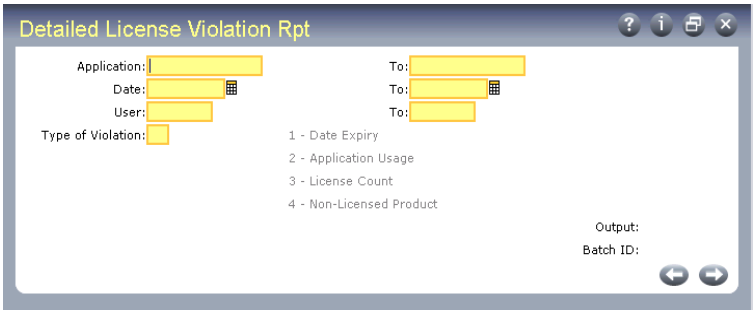


Fig. 8.12
Detailed License Violation Report (36.16.10.13)

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.

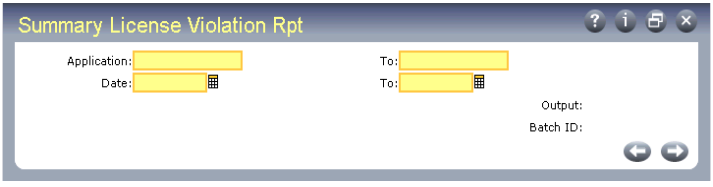


Fig. 8.13
Summary License Violation Report (36.16.10.14)

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

MFG/PRO 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

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

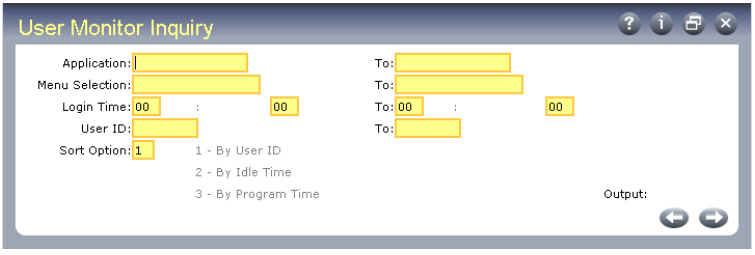
User Monitor Inquiry (36.16.12) displays users currently logged in, along with the:

- 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
- Client internet protocol (IP) address for Desktop user interface

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. 8.14
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 MFG/PRO 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) of Manager Functions helps you create and maintain multiple time zones.

The main features of time zones include:

- Define and maintain multiple time zones, including the changes required by daylight savings time.
- Load sample time zone data.
- Display and report time zone information.

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

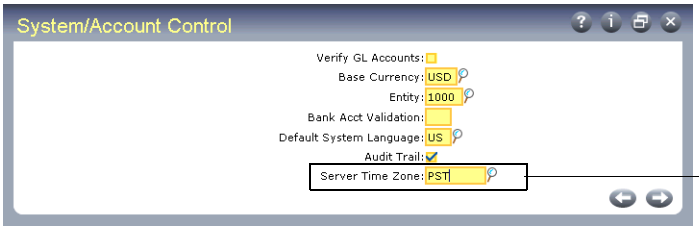
See *User Guide Volume 8A: Service/Support Management*.

Management Control. 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 “Defining Users” on page 42.

The system uses the date and time provided by the Server Time Zone field as the default value in date and time fields. This time zone defaults to the Time Zone field in User Maintenance (36.3.18).

Fig. 8.15 System/Account Control (36.1)



When MTZ is activated through SSM, the server time zone is set in both Service Management Control (11.24) and System/Account Control (36.1). If you try to change the server time zone when it is set from SSM, an error message displays.

If you are not using SSM, set the server time zone in the Server Time Zone field in System/Account Control, illustrated in Figure 8.15.

MTZ Setup Programs

Time zone setup programs are under the Multiple Time Zones Setup Menu (36.16.22) and include the following:

- Multiple Time Zones Maintenance (36.16.22.1)
- Multiple Time Zones Inquiry (36.16.22.2)
- Multiple Time Zones Report (36.16.22.3)
- Multiple Time Zones Load Utility (36.16.22.13)

It is best to use menu-level security on these programs, with the possible exception of the report and inquiry. Do not change time zone information without carefully evaluating the impact.

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). Figure 8.16 illustrates Multiple Time Zones Maintenance.

Fig. 8.16 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.

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

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.

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.

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

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.

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

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.

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

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.

Figure 8.17 illustrates the Multiple Time Zone Load Utility.

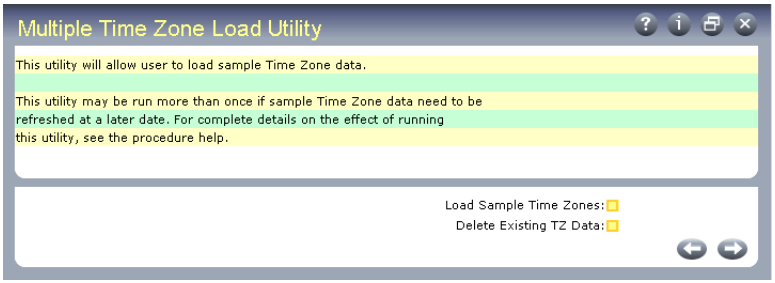
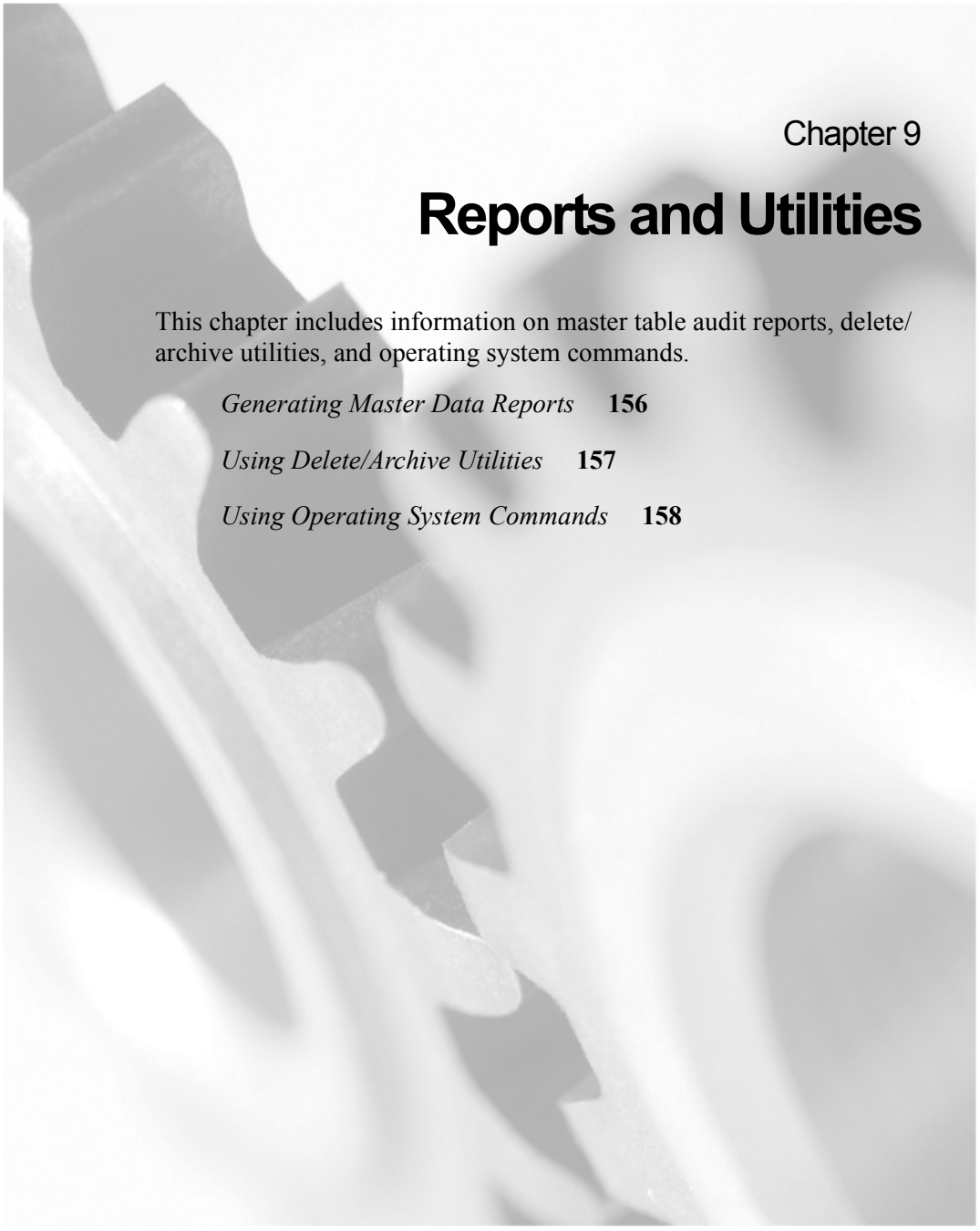


Fig. 8.17
Multiple Time Zone Load Utility (36.16.22.13)

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.



Chapter 9

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.

See *User Guide Volume 4A: Financials.*

To maintain detailed information for a critical subset of master tables, set Audit Trail to Yes in System/Account Control (36.1). Table 9.1 lists the database tables that the system tracks:

Table 9.1
Audited Tables

Table	Description	Table	Description
ac_mstr	Accounts	pac_mstr	Purchase Approval Codes
ad_mstr	Addresses	pc_mstr	Price Lists
bk_mstr	Banks	pld_det	Product Line Detail Accounts
bom_mstr	Bill of Material	plsd_det	Sales Accounts
cm_mstr	Customers	pl_mstr	Product Lines
cp_mstr	Customer Items	ps_mstr	Product Structures
ct_mstr	Credit Terms	ptp_mstr	Item Planning Data
cu_mstr	Currency	pt_mstr	Item Data
dpt_mstr	Department	spd_det	Account Validation
exr_rate	Exchange Rate	tax_mstr	Tax Data
flpw_mstr	Field Security	tx2_mstr	GTM Tax Rates
is_mstr	Inventory Status	usr_mstr	Users

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 Data Audit Detail Report (36.17.2) to show details about audited changes when Audit Trail is Yes in System/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 29.

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 MFG/PRO 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.

See “Audit Detail Delete/Archive” on page 128 for an exact procedure.

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

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 MFG/PRO. Use them as a convenient way of viewing and manipulating information.

- Use Exit to Operating System (36.24.1) to invoke a UNIX or NT session. To return to MFG/PRO, enter Exit.
- Use Program Execute (36.24.3) to run a Progress program. If the program is not in the current directory, specify the path.
- Use Program/Text File Display (36.24.4) 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.
- Use Disk Space Inquiry (36.24.13) to execute an operating system command to display statistics regarding the current database file size.

Tip
Add this function to the User Menu so that users can generate reports to a file and quickly review the content.

Chapter 10

System Cross-Reference

MFG/PRO's System Cross-Reference 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 MFG/PRO, 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

MFG/PRO 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 MFG/PRO-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 10.1 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.

Program Name	Description
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 10.2
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.

Program Name	Description
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.
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 11

Application Server

This chapter includes information on setting up application server definitions used with the Progress AppServer.

Progress AppServer

See the Progress documentation for more information on setting up and using AppServers.

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 *User Guide Volume 3: Manufacturing* for information on MRP.

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 "Example: Using an AppServer to Run MRP" on page 170.

As an example of how an AppServer can be used, this chapter includes instructions for setting up an AppServer to support enhanced MRP performance.

Before MFG/PRO 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 MFG/PRO 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 11.1 includes the data you would enter to define an AppServer used to improve MRP performance.

See page 170.

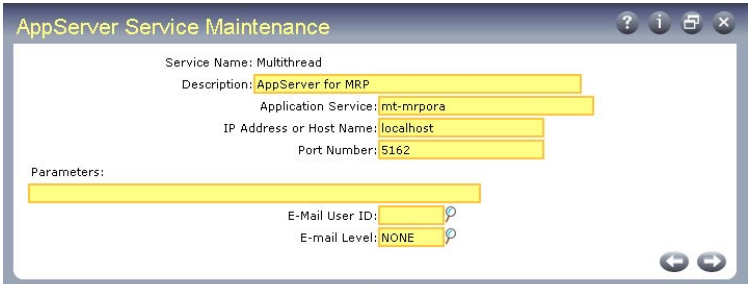


Fig. 11.1
AppServer Service Maintenance (36.19.1)

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 MFG/PRO, 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

See *User Guide Volume 3: Manufacturing*.

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.

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.

Tip
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 MFG/PRO Installations

Use this procedure to configure an AppServer instance for all MFG/PRO installations. If you have an MFG/PRO Oracle installation, additional configuration tasks are required.

See “Additional Oracle Tasks” on page 173.

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.

Tip
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=-db /dr05/mfgpro/pro/eb2/db/mfgprod -ld qaddb
    -znotrim -trig triggers -db /dr05/mfgpro/pro/eb2/db/hlpprod -ld
    qadhelp -db /dr05/mfgpro/pro/eb2/db/admprod -ld qadadm -d mdy
    -yy 1920 -Bt 3500 -c 30 -D 100 -mmax 6000 -nb 200 -s 63 -noshvarfix
  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

[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=-db /dr05/mfgpro/eb2/db/oraprod -RO -znotrim
    -trig triggers -db /dr05/mfgpro/eb2/db/mrp -dt ORACLE -U qad
    -P qad -c 250  -d mdy -yy 1920 -Bt 350 -c 30 -D 100 -mmax 3000
    -nb 200 -s 63 -noshvarfix
  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 MFG/PRO 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 MFG/PRO 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 MFG/PRO, you must use AppServer Service Maintenance (36.19.1) to define an application server connection master record.

See “Defining the AppServer” on page 169.

Additional Oracle Tasks

If you have an MFG/PRO 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
```

Where:

See the Progress AppServer documentation.

- */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

Starting and Stopping the AppServers

Tip
Click Start|Programs|Command Prompt to launch a DOS window.

The AppServer instance configured in the example on page 170 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.

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 171, 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 12

User Interface Management

This chapter discusses programs that let you modify the ways users interact with MFG/PRO 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 12.1 lists the user interface manager functions that are described in this chapter.

Table 12.1
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

See *User Guide: QAD Desktop* for details.

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).

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 18.

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:

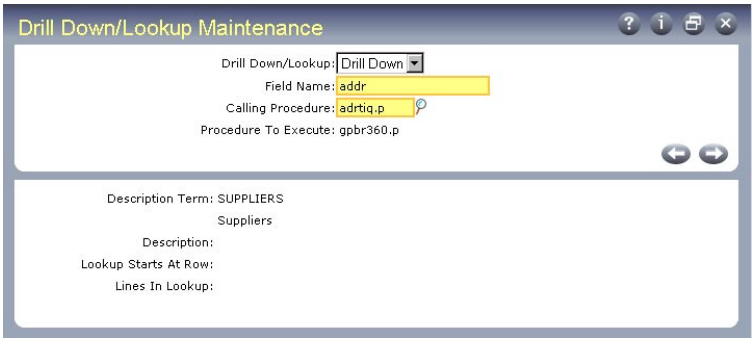
- The name of the field where you want the browse to display.
- The name of the program using the field.
- The program name of the browse to attach. If a lookup is missing for a particular field but exists for a similar one, use Lookup Browse (36.20.3) to determine the program that displays appropriate field values. Then use Drill Down/Lookup Maintenance to specify the same program for the similar field.

See “Creating Browses” on page 186 for details on creating browses.

Determining the name of the program and field depends on the user interface.

- In the Windows interface, run the program. Then select About… from the Help menu. The dialog box gives you the program name. Return to the program and place your cursor in the field where you want to attach the browse. Press Ctrl+F and note the field name that displays in the message.
- In the character interface, run the program. Note the program name in the upper left corner of the screen. Then place your cursor in the field where you want to attach the browse. Press Ctrl+F and note the field name.
- 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. 12.1
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 186.

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.

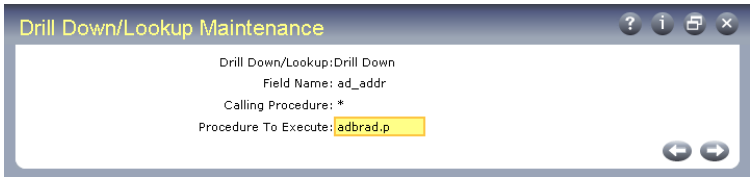
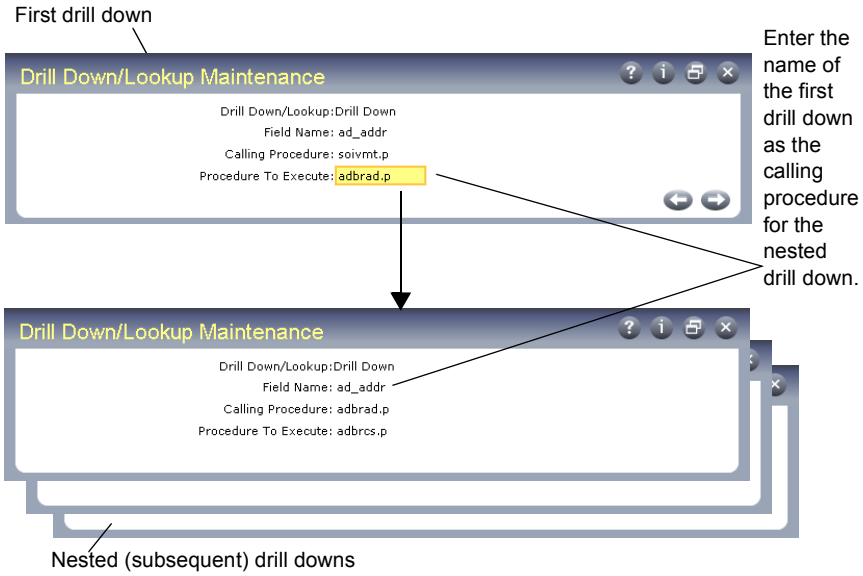


Fig. 12.2 Wildcards in Drill Down/Lookup Maintenance

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. 12.3
Nested Drill Downs



Planning for Upgrades

When you update MFG/PRO with 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.

See “User Menu and Function Keys” on page 68.

You can assign images to the buttons to make them easy to identify or use a text label only.

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.

See *User Guide: QAD Desktop* for details on adding links.

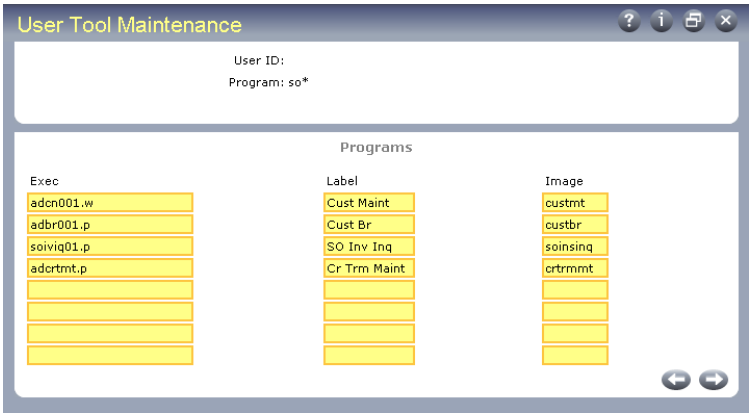
Images do not apply to Desktop. The link displays the text label specified. If no label is specified, the standard menu description from Menu System Maintenance is used.

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.

User Tool Maintenance

Figure 12.4 illustrates the User Tool Maintenance screen.

Fig. 12.4
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 68.

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.

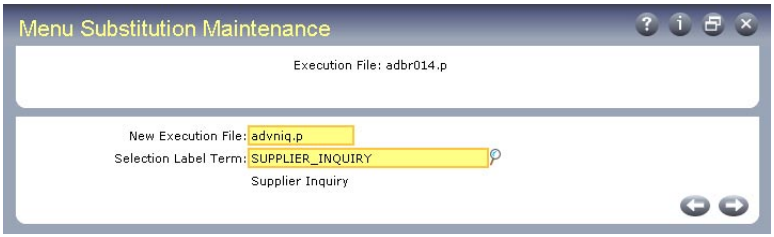
In the Windows and character interfaces, which program is invoked for a particular user depends on whether they have enabled menu substitution. Each user can turn menu substitution on or off in User Maintenance (36.3.18) or, in the Windows environment, from the Options menu.

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 MFG/PRO programs in these ways:

- Replaces browses with inquiry programs
- Replaces standard programs with custom versions

Fig. 12.5
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.

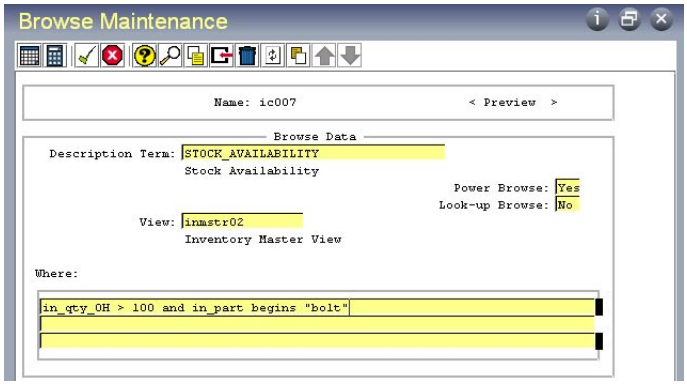


Fig. 12.6 Browse Maintenance (36.20.13)

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.

Tip Use the existing MFG/PRO 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.

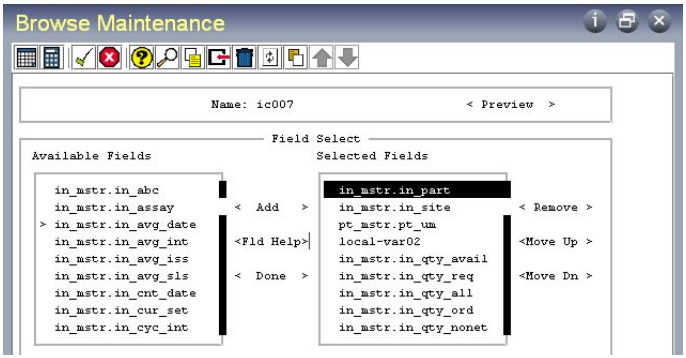
See “Creating Views” on page 190.

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 12.6 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. 12.7
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.

Tip
The Sort Columns field is enabled only for look-up browses.

12 Press Go.

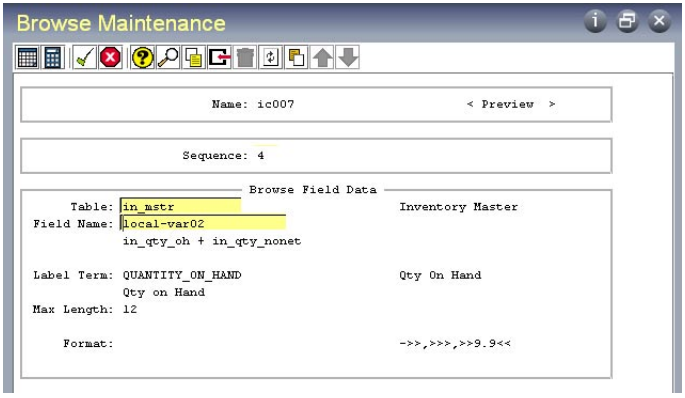


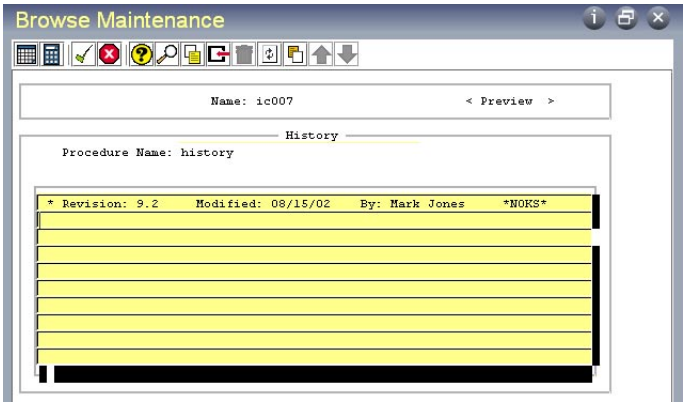
Fig. 12.8
Browse Maintenance, Browse Field Data

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. 12.9
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 MFG/PRO 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 12.10 illustrates how to create a view of selected fields from two tables.

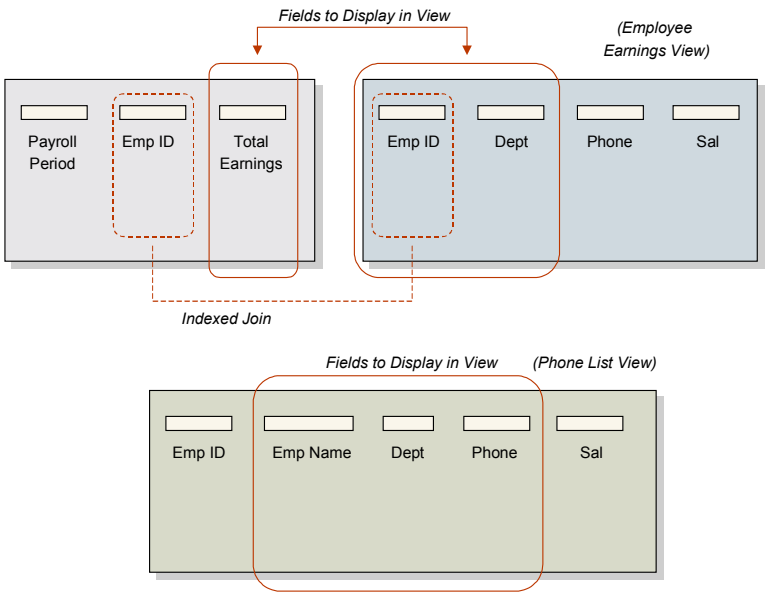
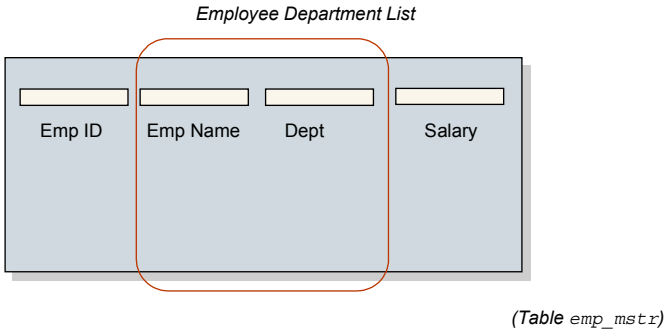


Fig. 12.10 Creating a View by Joining Two Tables

Figure 12.11 illustrates how to create a view of selected fields from one table.

Fig. 12.11
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 12.12 illustrates View Maintenance (36.20.18).

Fig. 12.12
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.

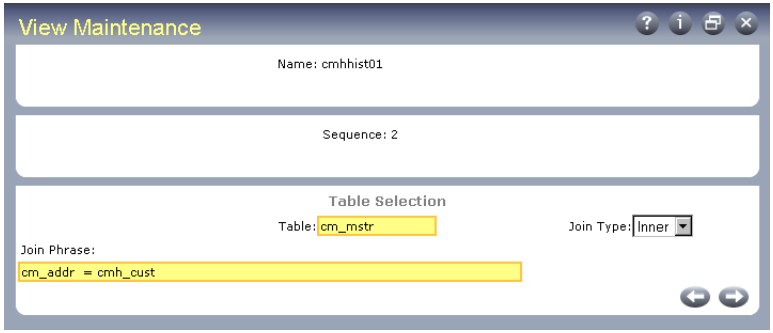


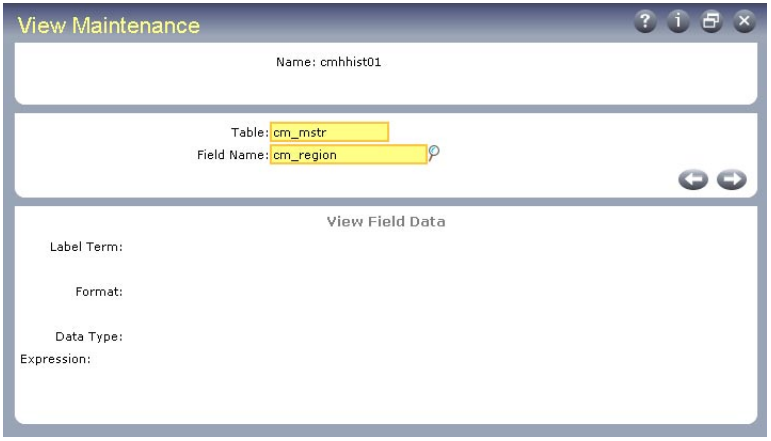
Fig. 12.13
View Maintenance, Table Selection

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 12.10). For a faster display of fields, use indexed fields in the Join Phrase.

9 Press End.

Fig. 12.14
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.

Tip
Search for a label term by entering a portion of a label, then use Next/ Previous to display available records.

11 If you entered a local variable in Field Name, enter its Label Term, Format, and Data Type.

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 MFG/PRO session if you attempt to use the view.

Index

N

O

P

V

W