MFG/PRO® eB

User Guide
Volume 9
MANAGER FUNCTIONS

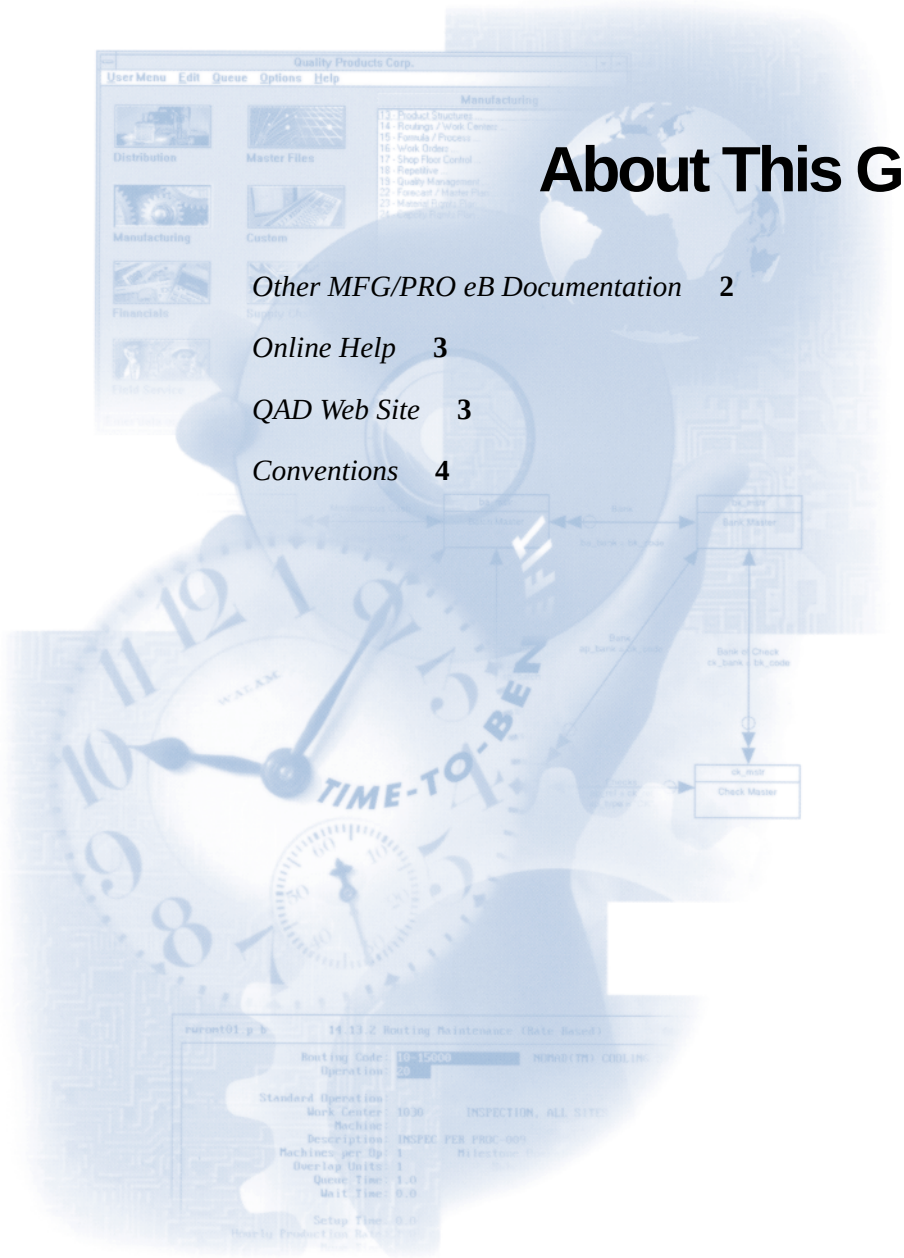
SYSTEM CONSTANTS
USERS AND SECURITY
SYSTEM INTERFACE
MULTIPLE DATABASES
PRINTERS AND BATCH PROCESSING
CIM INTERFACE
DATABASE MANAGEMENT
REPORTS AND UTILITIES
SYSTEM CROSS-REFERENCE
USER INTERFACE MANAGEMENT

78-0514A
Printed in the U.S.A.
August 2000

78-0514A

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 eB. Most of these programs are on the Manager Functions menu (36).

Other MFG/PRO eB 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 instructions on navigating the MFG/PRO eB Windows and character environments, refer to *User Guide Volume 1: Introduction*. Navigation information for the Network User Interface (NetUI) is provided in the *Network User Interface Guide*.
- For information on using MFG/PRO eB, refer to the *User Guides*.
- For information on using features that let MFG/PRO eB work with external applications, see the *External Interface Guides*. For example, these guides describe the Warehousing application program interface (API) and Q/LinQ, the tool set for building and using tools that perform complex data exchange between MFG/PRO eB and external systems.
- For technical details, refer to *File Relationships* and *Database Definitions.*
- To view documents online in PDF format, see the *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 eB installation guides are not included on *Documents on CD*. Printed copies are packaged with your software. Electronic copies of the latest versions are available on QAD's Web site.

Online Help

MFG/PRO eB 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 eB environments, refer to *User Guide Volume 1: Introduction* and the *Network User Interface Guide*.

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 eB users with a QAD Web account, product documentation is available for viewing or downloading at:

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

To obtain a QAD Web account, go to:

http://support.qad.com/

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 eB 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 eB.

Conventions

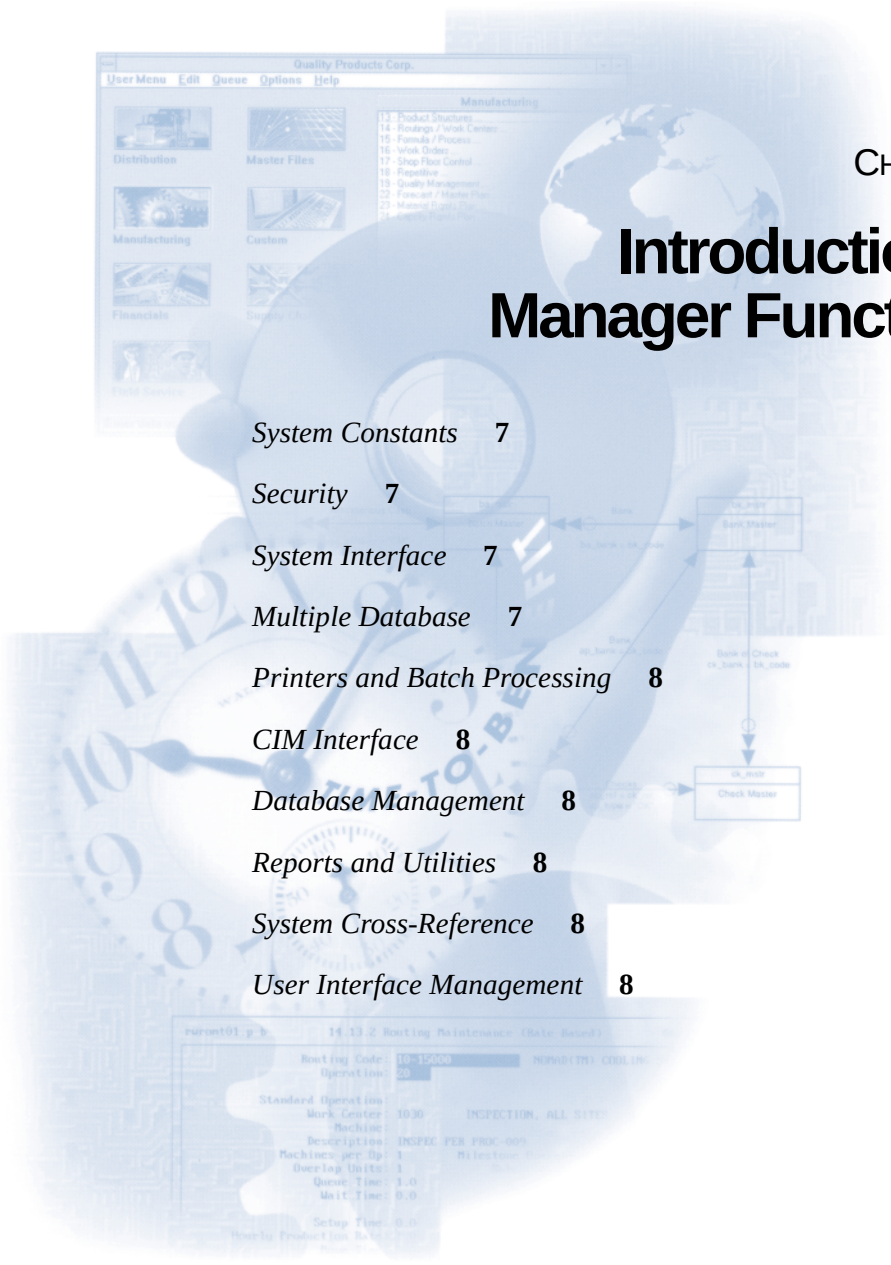
MFG/PRO eB is available in several interfaces: Windows, character, Web browser, and an interface for object-oriented programs. To standardize presentation, the documentation uses the following conventions:

- MFG/PRO eB screen captures show the Windows interface.
- References to keyboard commands are generic. For example, choose Go refers to F2 in the Windows interface and to F1 in the character interface. Throughout MFG/PRO eB, the PROGRESS status line at the bottom of the program window lists the main UI-specific keyboard commands used in that program.

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

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`	Indicates 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.

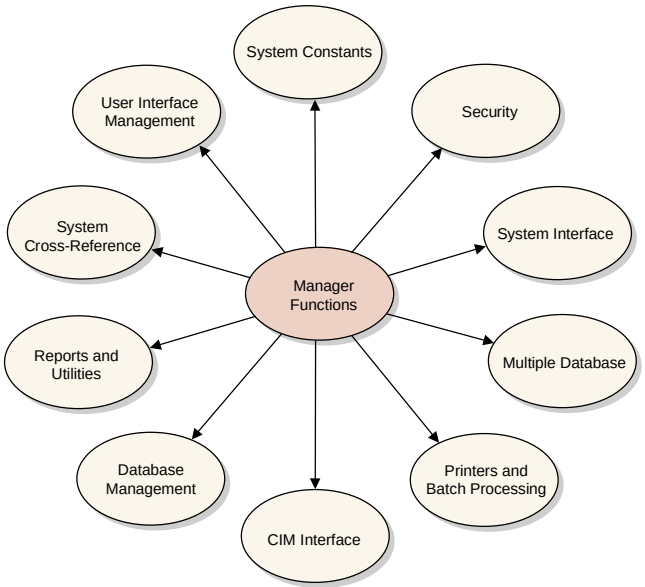


CHAPTER 1

Introduction to Manager Functions

Manager Functions includes tasks typically performed by system administrators within MFG/PRO eB. 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:

- The System/Account Control File (36.1) affects processes throughout MFG/PRO eB. 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 NetUI Security Menu (36.3.21) are discussed in the *Network User Interface Guide.*
- 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 the *External Interface Guides*.
- AppServer Menu (36.19) is discussed in *External Interface Guide: QAD Storefront.*

- User interface functions found on the NetUI Menu (36.20.10) are discussed in the *Network User Interface Guide* and the *Network User Interface Installation Guide*.

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, rounding methods, and number range management (NRM), which supports regulatory controlled document numbering. NRM includes the content and sequencing of a numeric series, as well as preventing gaps in a series.

See Chapter 2, “System Constants,” for details.

Security

MFG/PRO eB offers several levels of security, including database, menu, field, entity, site, account, and inventory movement code. You can implement these levels by user ID or user group, password, or both.

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

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.

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

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. To schedule batch jobs at the operating system level, see the appropriate *System Administration Guide.*

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 eB database. Using CIM, data can be added using standard program validation.

Database Management

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

MFG/PRO eB 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 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 maintain field, program, and file relationships in your database. If you customize MFG/PRO eB, this is an essential set of programs.

User Interface Management

See Chapter 11, “User Interface Management,” for details.

The UI: Manager Functions menu provides several programs used to customize such user interface elements as browses and to set user preferences for such things as menu styles and tool bar buttons.

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 eB. Table 2.1 lists available functions.

Table 2.1
System Constants Menu (36.2)

Number	Menu Label	Program
36.2.1	Holiday Maintenance	mgcn026.w
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	adcn020.w
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	mgcn021.w
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

Maintaining Holiday and Shop Calendars

The shop calendar is required for the 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.

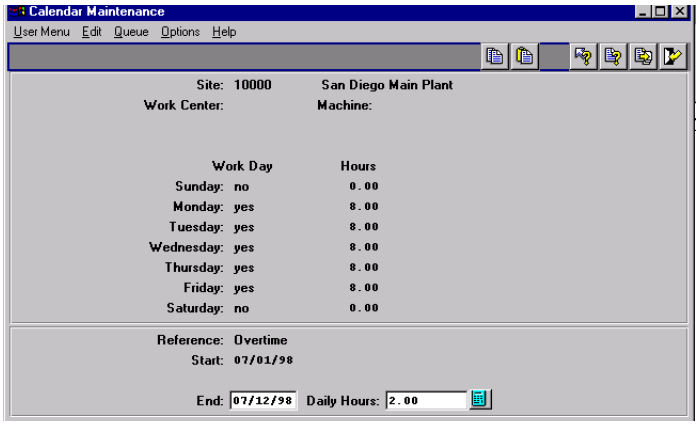


Fig. 2.1
Calendar Maintenance (36.2.5)

In a calendar, work days are flagged Yes and nonwork days 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.

See *User Guide Volume 7: Release Management* for details.

Enter supplier or customer address codes as the site code to set up calendars for suppliers and customers. These calendars support ship-from and ship-to calendars used in Release Management.

Holiday Maintenance

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

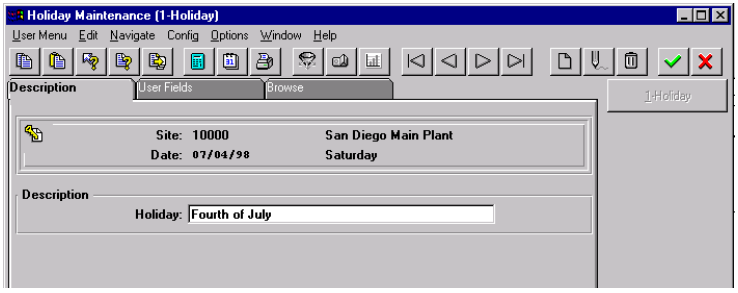


Fig. 2.2 Holiday Maintenance (36.2.1)

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 eB 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).

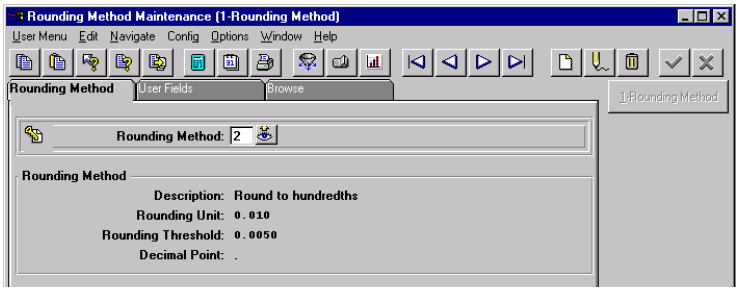


Fig. 2.3 Rounding Method Maintenance (36.2.9)

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.

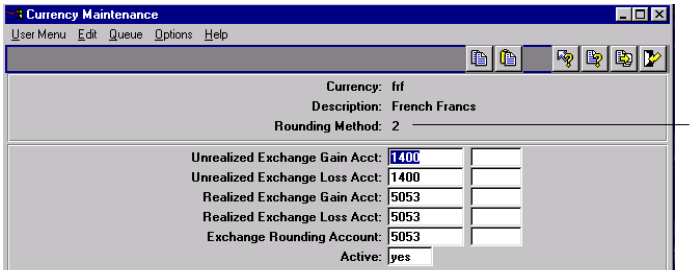
Rounding Threshold. Enter the number at which monetary values are rounded up. This number must be less than the number entered for the rounding unit.

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 eB 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 help windows.

Before implementing a module or section of MFG/PRO eB, the implementation team should determine which fields should have generalized codes and help windows.

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 Window Help Maintenance (36.4.21).
- 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 17.

Field Validation

Before entering a list of generalized codes for a field, you must know the field's name and size. 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.

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

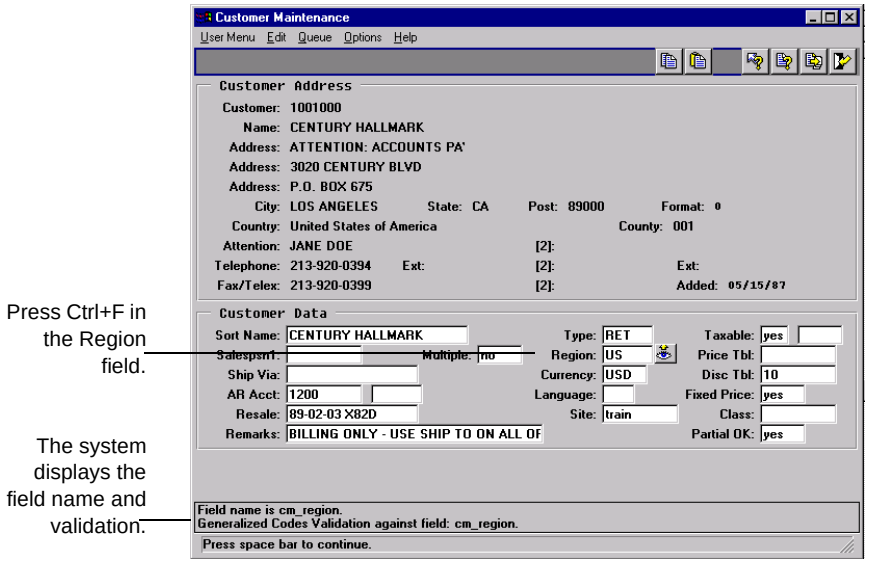


Fig. 2.5
Testing Validation of Region Field

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 2.6 illustrates Generalized Codes Maintenance.

Fig. 2.6
Generalized Codes Maintenance (36.2.13)

Use the field name to enter a list of values and comments. Values cannot exceed the length of the field.

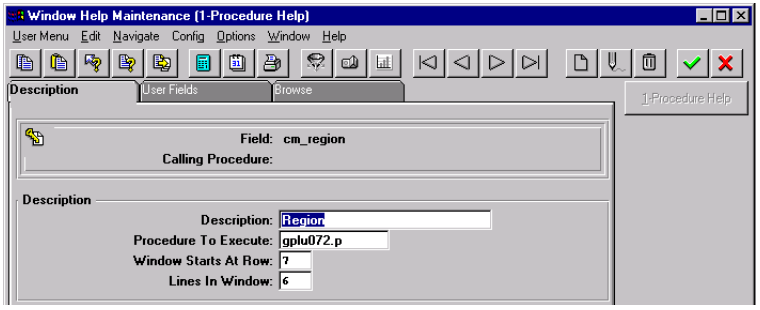
Adding a Help Window

To set up a help window to access generalized codes, use Window Help Maintenance (36.4.21). Enter the field name and the procedure to execute.

- For a character-based scrolling window, enter `swcode.p`.
- For a Windows-based browse, enter `gplu072.p`.

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

Fig. 2.7
Window Help Maintenance (36.4.21)



Note that the description defaults from the data dictionary, but can be changed here. If no description exists, the field name is a local variable.

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*. 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.

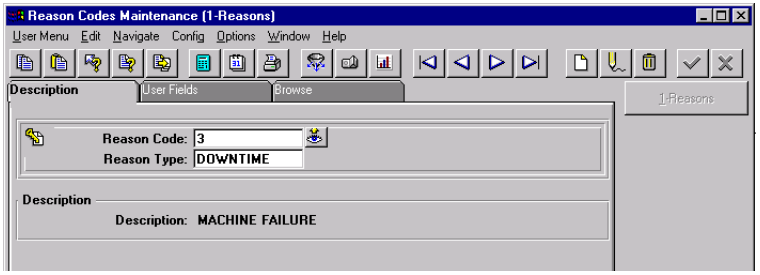
Important If you change the data dictionary, keep careful records and be prepared to repeat the change when new versions of MFG/PRO eB that update the data dictionary are installed.

Using Reason Codes

Reason codes are used in sales quotes, shop floor reporting, and the Product Change Control (PCC) module. Add other custom uses as needed.

- Use codes of type QUOTE in the Reason Lost field of sales quotations.
- Use codes of type DOWN or DOWNTIME in the Reason field for labor feedback programs (17.1–17.4).
- Generate reports on downtime organized by reason code using the Downtime by Reason Report (17.17).
- Use REJECT and REWORK codes in shop floor programs for reporting quantity variances.
- Codes used in the PCC module are user-defined. They specify severity levels related to approval of change documents.

Fig. 2.8
Reason Codes Maintenance (36.2.17)



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 2.9 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 (06:15:96).

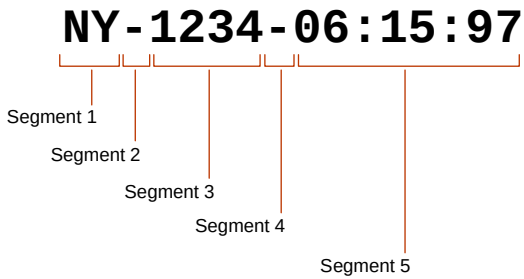


Fig. 2.9
Example Sequence Number

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, 06:15:97 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 at the request of a client program. 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 let a client program pass a sequence number to NRM for validation 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 2.10 illustrates the life cycle of a sequence.

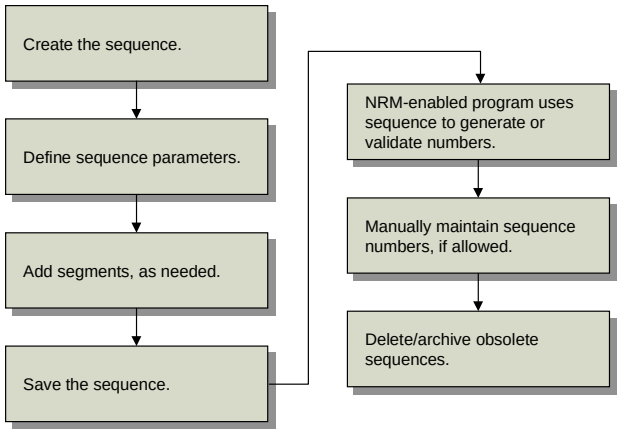


Fig. 2.10
Sequence Life Cycle

To set up a sequence, create an ID, define general parameters, and add appropriate segments. Once a sequence is defined, an MFG/PRO eB 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 archived unneeded sequences.

NRM Sequences in MFG/PRO eB

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

Shipping

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

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

General Ledger Daybooks

See *User Guide Volume 4A: Financials* for details.

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

Fixed Assets

See *User Guide Volume 4B: Financials* for details.

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

WIP Lot Trace

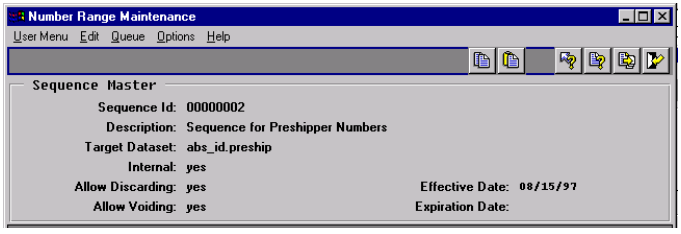
For details, see the *PRO/PLUS User Guide.*

An optional NRM sequence number can be specified in the WIP Lot Trace Control File (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. 2.11
Number Range Maintenance (36.2.21.1)



Sequence ID. Enter a code uniquely identifying a sequence. Create a new sequence or use the Up and Down arrows 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.

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.

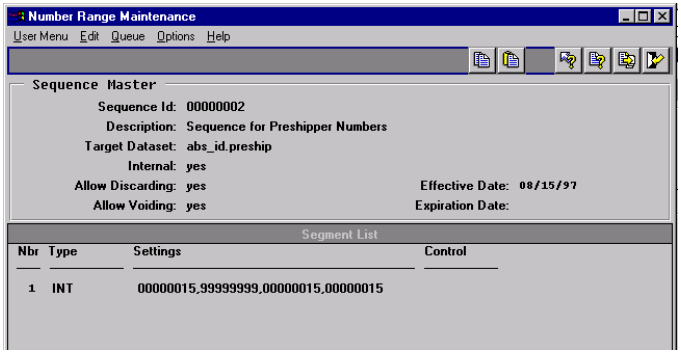
Effective Date. Indicates the earliest date when this sequence can be used.

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. 2.12 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. 2.13 Fixed Segment Editor

Integer Segment Editor

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

Fig. 2.14 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 07/96, 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.

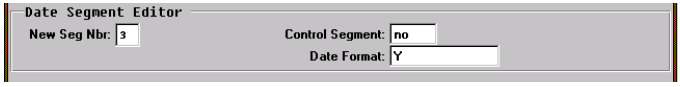


Fig. 2.15 Date Segment Editor

Fiscal Segment Editor

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

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

Fig. 2.16 Fiscal Segment Editor

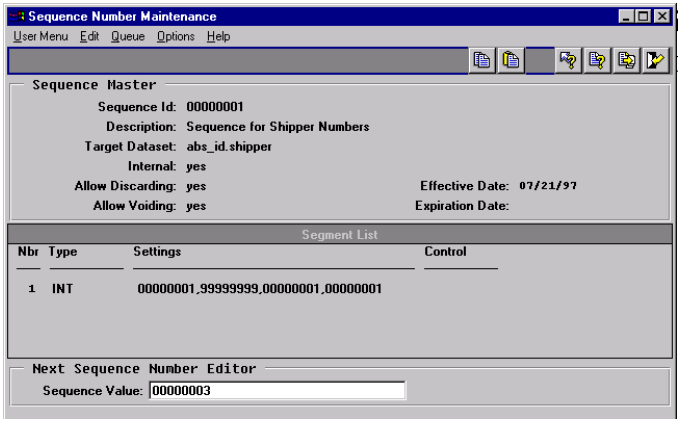
Setting Sequence Values

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. 2.17
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.

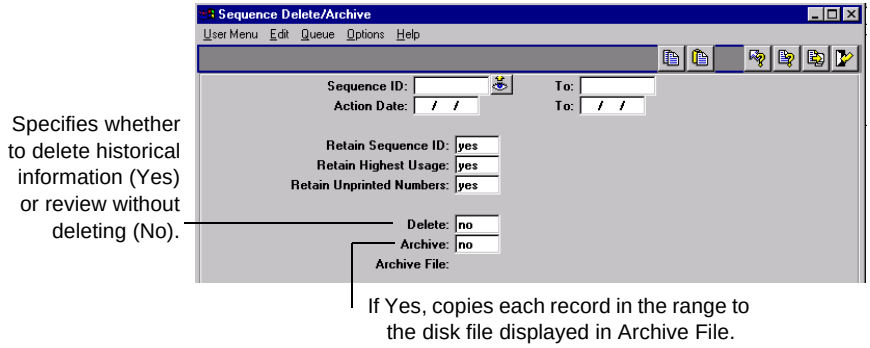


Fig. 2.18
Sequence Delete/ Archive (36.2.21.23)

CHAPTER 3

Users and Security

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

Security Overview

MFG/PRO eB provides options for security on several levels.

- Database security determines whether a user can log into a database. This level of security is defined in the Security Control File (36.3.24), and affects how some of the other types of security operate.
- 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 the *Network User Interface Guide* for details.

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

See page 134 for details.

- Master file audits (36.17 menu) show when and by whom changes were made to any master file.

See "Assigning Security to OBCM Programs" on page 153.

- Program, window, and field security for Object-Based Component Model (OBCM) programs can be defined in Window Security Maintenance.

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

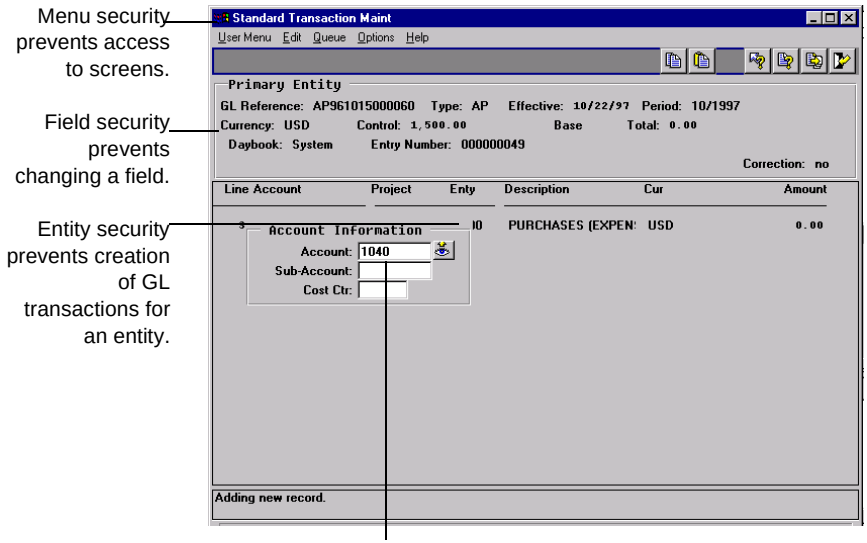


Fig. 3.1
Types of Security

By default, no security is defined in MFG/PRO eB, and all users have free access to the system. 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 character strings.

- For entity and field security, the character string is a user ID or password.
- For menu security, site security, inventory movement code security, and GL account security, the character string can be a password, user ID, or user group.

When a user tries to do something that is controlled by security, the system compares the string or strings defined in the security records with the values 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 password, a 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 Password Maintenance	mgpwmt.p
36.3.2	Menu Password Change	mgpwcg.p
36.3.4	Field Security Maintenance	mgflpwmt.p
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
36.3.13	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
36.3.18	User Maintenance	mgcn023.w
36.3.19	User Browse	mgbr010.p
36.3.20	User Password Maintenance	mgurmtp.p
36.3.24	Security Control File	mgurpmmt.p

Setting Up the Security Control File

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

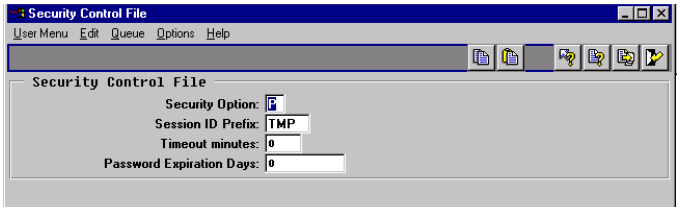


Fig. 3.2
Security Control File (36.3.24)

Security Option. Select a database security option. Valid options are password (P), user ID (U), or both (B).

See “Selecting a Security Option” on page 34.

Session ID Prefix. Enter a prefix for temporary system-generated work files. These are created in the directory where the user started MFG/PRO eB. Defaults to TMP. Modify it 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 displays the date the user last modified the password.

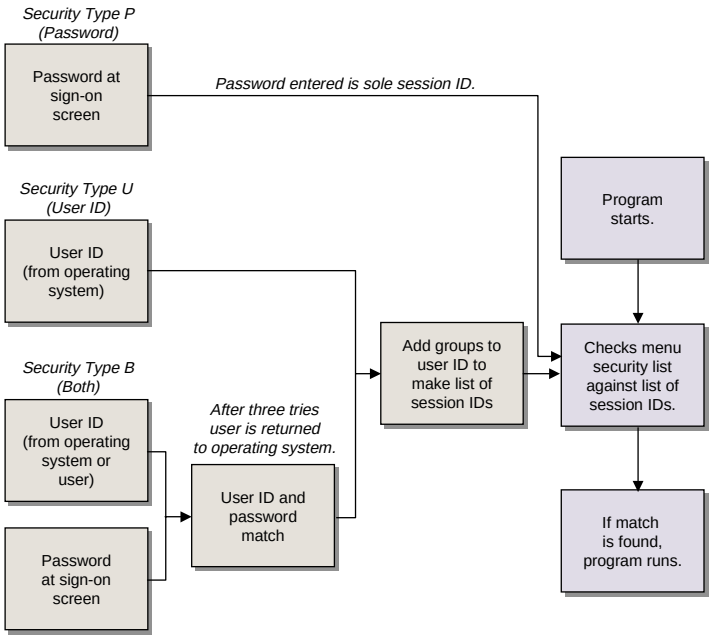
Selecting a Security Option

Select a security option in the Security Control File to determine what information a user must supply before he or she is allowed to log into the database. The three security options also affect how other security functions are applied.

- Password (option P)
- User ID (option U)
- Both password and user ID (option B)

Figure 3.3 illustrates how the three security options affect what happens when a user tries to log on to a database.

Fig. 3.3
Security Options



Password Security

Password security (option P) is the default for a new database. When this option is active, each user must enter a password at the sign-on screen. The password is passed directly to each program invoked. Define user passwords in User Maintenance (36.3.18).

With this option, security must be provided through other security layers. Option P does not prevent anyone from logging into a database. It is only as good as the individual menu, account, field, or entity security set up in the database. For this reason, most companies do not find option P a realistic alternative. It is used mainly for demonstration purposes or test databases, when critical data is not affected.

User Security

When user security (option U) is active, no information is required when a user logs in. The system captures the user ID from the operating system. If that user has been set up in the database, login is successful. Otherwise, login fails.

Once in the database, what the user can do is determined by other types of security, as well as which groups the user belongs to. Groups provide a powerful tool for setting up access control.

See “Defining Users and User Groups” on page 38 for details.

Both User and Password Security

When option B (both) is active, a user must enter both a user ID and a password. 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.

This option combines password restriction for database access with user groups for defining other security features. Option B is the most robust security option.

Warning If you select option B, you must set up at least one user ID with a valid password in User Maintenance before you exit the system. Otherwise, reentering the system will be difficult.

If this happens and you have full PROGRESS, reset the value with the following commands:

```
find qad_wkfl where qad_key1 =
“usr_mstr:control” and qad_key2 = “”.
qad_charfld[1] = “P”.
```

Option B offers two advantages:

- You can effectively separate MFG/PRO eB security from the operating system’s security. The user ID in MFG/PRO eB does not have to be the same as the user ID referenced by UNIX or NT.
- You 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 eB 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 eB user, anyone with access to the root user ID at the operating system level can get access to any information in the MFG/PRO eB databases.

PROGRESS Security

If necessary, you can use PROGRESS utilities to place additional control over how a user’s ID is set. Do this by accessing 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) file.

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

1. No PROGRESS users are defined and the –U and –P options are not specified. This is the default. The PROGRESS user ID is set to the operating system login or the network login 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 start-up.
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) file. 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 eB provides an option to save the user ID for the next login when either security option B or P is active.

Note For security option U, saving the logon user ID is disabled, since the system always uses the network logon ID.

If you need to log into MFG/PRO eB 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 and User Groups

The groups a user belongs to can determine whether a user is given access to menus, fields, and sites. Each program is always passed the user's ID and any group names associated with the user. Define users in User Maintenance (36.3.18).

Language

See "Using Multiple Languages" on page 54.

In a multi-language environment, the system displays the correct menus, messages, and other interface features based on the user's associated language.

Password

System administrators typically set up passwords for users in User Maintenance. If you want to let users change their own passwords but not other attributes, give them access to User Password Maintenance (36.3.20). Passwords expire based on the value of Password Expiration Days in the Security Control File.

E-mail Address and Definition

See "Building an E-Mail System Interface" on page 64.

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

User Profile

For an example, refer to "User Group Example" on page 42.

The user profile includes a name that displays on reports and browses. The User Type can be either Customer and Employee. Specify Customer if the user specified is defined in Customer Maintenance (2.1.1).

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.

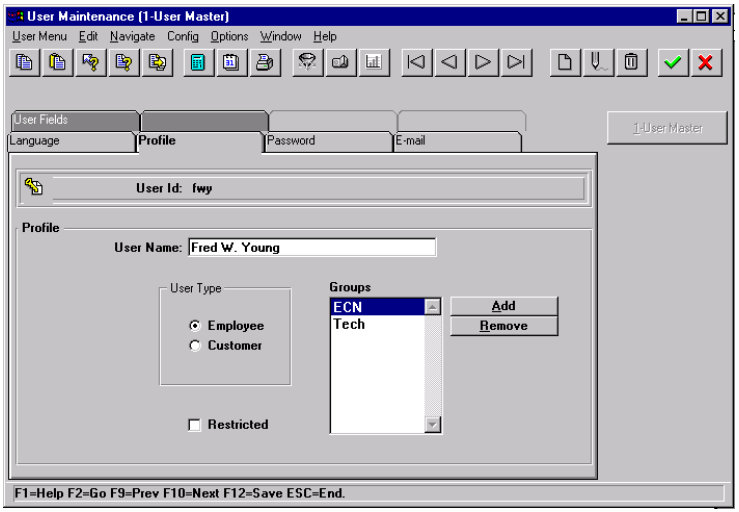


Fig. 3.4
User Maintenance (36.3.18), Profile Folder

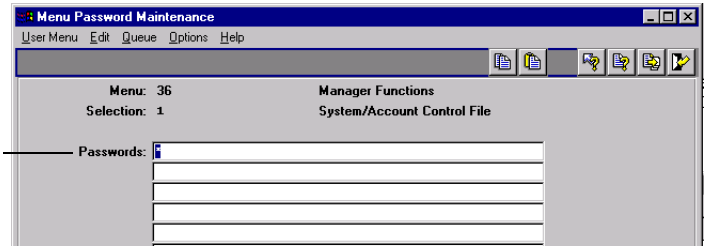
If security option U (user) or B (both user and password) is active, the system administrator should assign each user to groups, since this makes assigning menu security easier.

- 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 in the pop-up that displays when you choose the Add button. 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.

Using Menu Security

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

Fig. 3.5
Menu Password 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 Files

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 Password field.

- Use the asterisk to give access to all users and groups. A blank in the Groups field 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 security 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 the Network User Interface (NetUI), restricted menus do not display if Restrict Menu is set to Yes for the user in User Option Maintenance (36.20.10.1). If this setting is No, a restricted menu item is disabled and displays with light gray text. Users can choose not to see restricted menus by selecting Menu Hiding from the Options menu.

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 Password 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. Although you can use passwords (security option P), the easiest method is with user groups (security option U or B).

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.

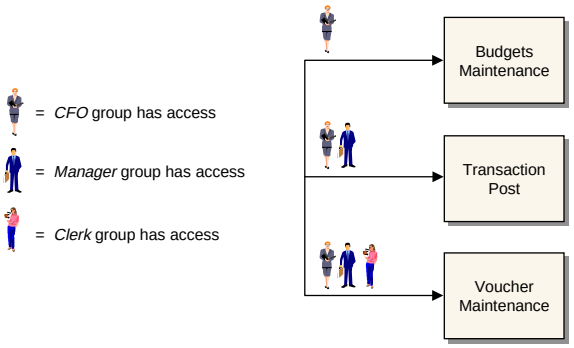


Fig. 3.6
Using Groups to Give Access

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 Password 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.

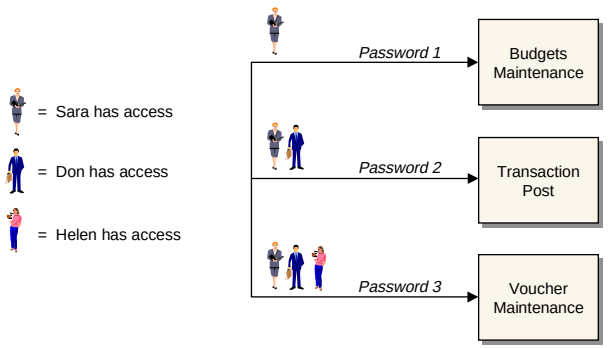
Security Example Without Groups

To set up several levels of access with security option P, you must create different passwords for each level of access. Then, you have two choices:

- Set up multiple passwords for each function that corresponds to several levels of access
- Require your users to log on with different passwords to get into different functions.

Example At Company A, Sara is CFO, Don is AP Manager, and Helen is an AP clerk. As Figure 3.7 shows, Sara needs three different passwords to gain entry to all the levels she is authorized to access.

Fig. 3.7
Using Passwords to Give Access



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 the security option in effect. If option P (password) is active, the user's password must match a value associated with the field. If option U (user ID) or B (both password or user ID), the user's ID is matched against values specified for the field.

See "Specifying Groups" on page 41.

User groups are supported through a two-step process.

Field Password Validation

In the standard release of MFG/PRO eB, security is not active for any fields, and only a few fields are eligible for field security. To find out whether a specific field is eligible, access the field on a screen and press Ctrl+F. For eligible fields, the message Password Validation displays. You can also use the Dictionary Field Security Report (36.3.6) to determine which fields can be given security.

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 or passwords 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.

Note You must have already set up user security (user IDs at login) prior to completing this task.

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 file and field name on which to activate field security. Once you enter a valid field and file 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 or passwords 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 the mgfldcmt.p utility 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.

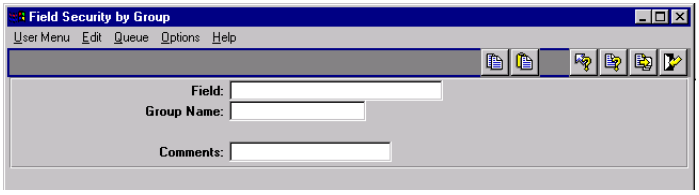


Fig. 3.8
Field Security by Group (36.3.7)

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.

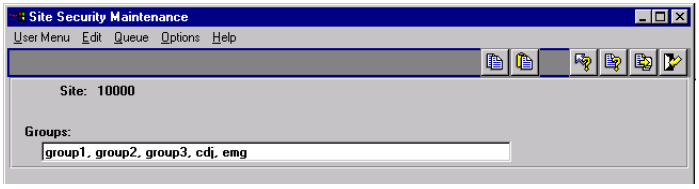
Once Field Security by Group is executed for a field and group, all users who belong to the group display in Field Security Maintenance (36.3.4) as authorized to access the field. The Comments field in Field Security by Group displays as the comment for the field and user combination in Field Security 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. Security option must be U or B to use site security.

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 site's Group field in Site Security Maintenance (36.3.16).

Fig. 3.9
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, Service Contract Control File (11.5.24), and Quality Management Control File (19.24) do not use site security.

- You must set up each database individually.

Implementing Site Security

Because of the complexities of MFG/PRO eB 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.

If site security is active and the security option is changed to P in the Security Control File (36.3.24), all users are restricted from making site security-controlled transactions.

To implement site security, first select security option U or B in the Security Control File. Then associate groups with users in User Maintenance. You can use previously defined groups to implement site security.

See “Specifying Groups” on page 41.

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

Tip
Entity security has no effect on transactions in other modules.

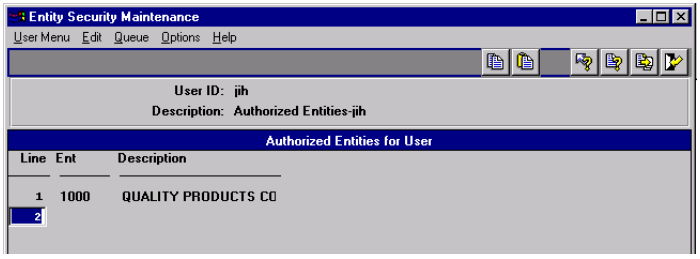
When entity security is in place, only authorized users can complete update transactions in the General Ledger and Fixed Assets modules for particular entities. Update transactions include:

- Transaction maintenance
- Posting
- Consolidation
- Export and import
- Budget maintenance
- Opening and closing fiscal periods

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 3.10.

Fig. 3.10
Entity Security Maintenance (36.3.13)



An asterisk in the Entity field indicates that a user can access 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 if:

- You are using security options U (user ID) or B (both user ID and password).
- Verify GL Accounts is Yes in the System/Account Control File.
- Account security groups are validated against groups assigned to users in User Maintenance.

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.

See “Specifying Groups” on page 41.

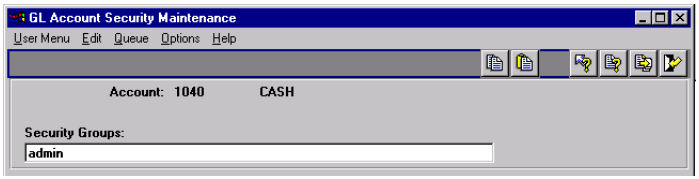


Fig. 3.11 GL Account Security (36.3.9)

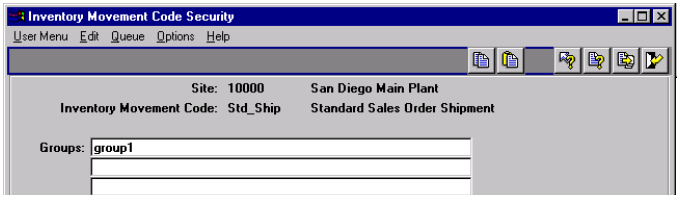
When a user attempts to create a transaction affecting an account, the system checks the list of groups associated with the user against the list associated with the account. If a match is not found, a message displays and the user cannot complete the transaction.

Account security is not applied during Transaction Post. Use Menu Password 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 user access to individuals and groups when using a specific inventory movement code at a particular site.

Fig. 3.12
Inventory Movement Code Security (36.3.11)



See *User Guide Volume 2: Distribution.*

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.

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

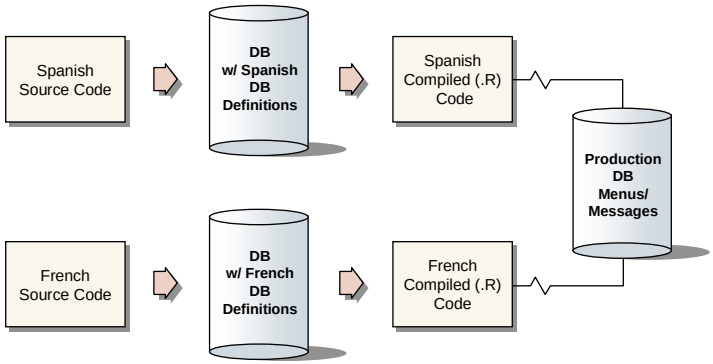
The System Interface menu contains programs that control menus, messages, multi-language installations, and help. If you are using QAD's Network User Interface (NetUI) for the Java platform, interface details are discussed in the *Network User Interface Guide*.

Using Multiple Languages

MFG/PRO eB 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 eB 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 code for each user in User Maintenance (36.3.18). This ensures that when the user logs on, MFG/PRO eB calls the PROGRESS programs for that person's language.

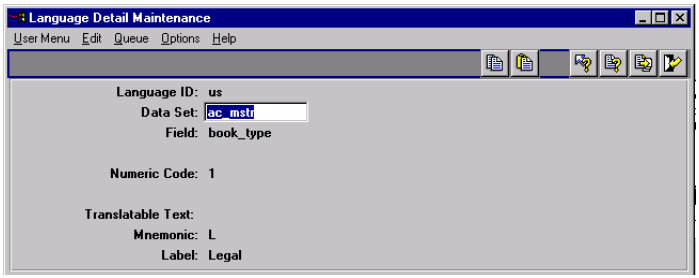
See "Defining Users and User Groups" on page 38.

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 eB 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 file name, or an abbreviation of the functionality for a field.

Field. Enter the field name associated with the data set. To find a field name, move the cursor to the field and press Ctrl+F.

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
- Password protect menus

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

All menu information is contained in a file called mnd_det. View its structure in the MFG/PRO eB Data Dictionary. With each release, you receive the latest version of this file, which you should load into your databases. As QAD develops new programs, it populates this file with new records and alters existing records. When loading the latest version of the file, 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 file 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.

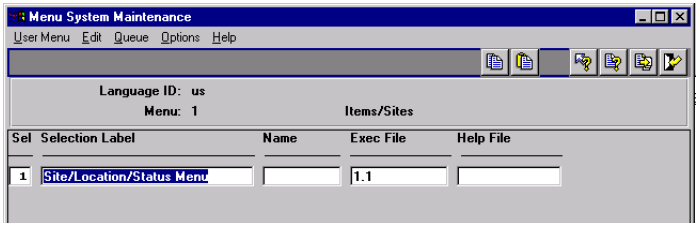


Fig. 4.4
Menu System Maintenance (36.4.4)

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.

User 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.

Once you assign a menu item to a function key for a particular user ID, the user has the following ways to execute that program:

- From a character environment, Press F6. A list of menu items appears. Choose the one you want by highlighting it and pressing Enter or Go.

  Note From a Windows environment, access user functions by choosing them from the User Menu.

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

For example, 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.

Note Do not use function keys or the function menu to access a maintenance screen. 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.

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.

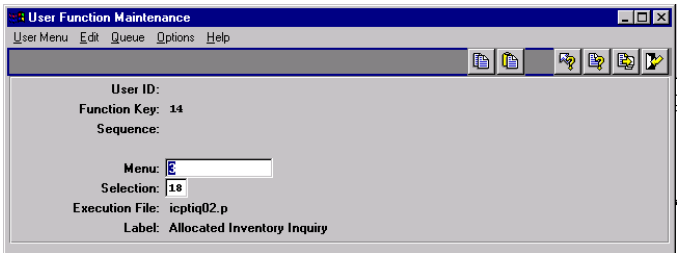


Fig. 4.5
User Function Maintenance (36.4.11)

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.

User Menu

Press F6 in a character environment to display a pop-up window listing all menu choices assigned to your user ID. There is no relationship between this order and the function key assigned, and the function key is not shown. Use the Tab or Pivot key to move between columns. You can use the F8 or F9 keys to move up and down, and search in either column.

Tip
Menus sort in lexical order so that 13 appears before 2 if you are in the Menu Selection column.

Press End to display all the menu choices assigned to the blank user ID. To select a menu item, highlight it and press Enter or Go.

Note In the Windows environment, click on the User Menu option in the menu bar and choose an item to execute.

Stored Values for Fields

F12 or Ctrl+A stores values for any number of fields. Values are pasted back when you press F12 or Ctrl+A again. The field values are stored separately for each field and for each user, and the values are saved between logon sessions.

Example Using Sales Order Maintenance, enter a value in Channel and press F12 while the cursor is still in the field to store the value. Then enter a value in Credit Terms and press F12 to store that entry. Finish entering the sales order. The next time you enter a sales order, press F12 in the empty Channel field and the stored value is entered. Press F12 on the Credit Terms to input the stored credit terms.

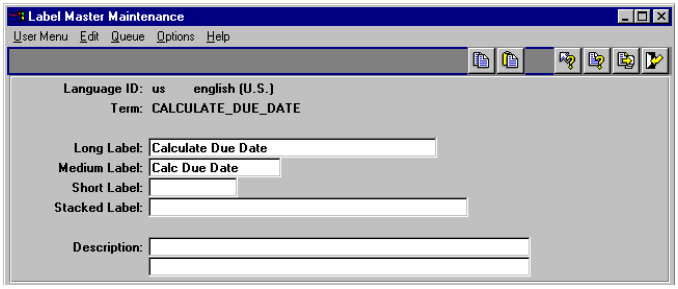
To store field values in a maintenance program, you must select Go or Enter through the frame containing the field values you stored. If you exit the frame using End, the field values you stored are not saved. This is only true in maintenance and other update programs. Values stored using F12 in reports and inquiries are saved even if you do not select Go or Enter to complete the frame.

Modifying Labels

MFG/PRO eB 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 the Label Control File (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 eB 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 eB, and an MFG/PRO eB program error message is substituted, so this should occur rarely.

You can modify MFG/PRO eB 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.

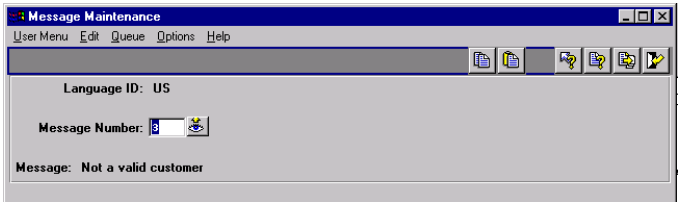


Fig. 4.7 Message Maintenance (36.4.7)

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 eB in a later version.

Using Field and Procedure Help

MFG/PRO eB 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.

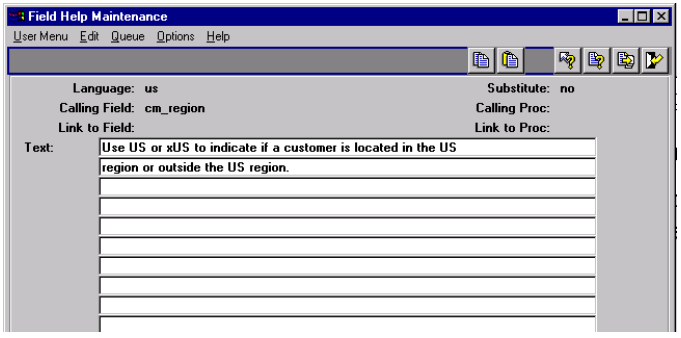
In the Windows interface, online help uses the standard Windows help format (.hlp files). In addition to receiving context-sensitive help on the active field and procedure, you can also access other help topics through the help menu, hypertext links, and keyword searches.

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 eB.

Fig. 4.8
Field Help Maintenance (36.4.13)



See “Setting Up User Interface Profiles” on page 150.

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 Interface Profile (36.20.4).

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.

Setting Up Window Help

Use Window Help Maintenance (36.4.21) to specify the program that executes when a user selects Help from a particular field. The most common use is to display generalized codes.

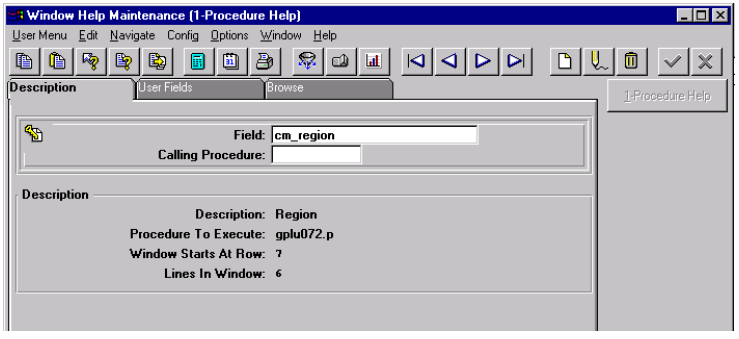


Fig. 4.9 Window Help Maintenance (36.4.21)

You can also set up window help for any field that acts as an index to a maintenance screen. Often, however, the search function must be written as a custom program.

If window help is missing for a particular field but exists for a similar one, use Window Help Browse (36.4.22) to determine the program that displays appropriate field values. Then use Window Help Maintenance to specify the program for the similar field.

When you update MFG/PRO eB with a new version, be careful when loading flh_mstr. This file contains the records created by Window Help Maintenance. If you have customized it, make sure that the new version does not overwrite your customization.

Most programs attached to a function with Window Help Maintenance look up values in some file. 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.

Building an E-Mail System Interface

Some functions can be configured to send E-mail messages to designated users. For example, E-mail messaging is used in both the Product Change Control module and the Global Requisition System.

To take advantage of this feature, the E-mail system must be defined and addresses specified. The MFG/PRO eB 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 eB. 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.

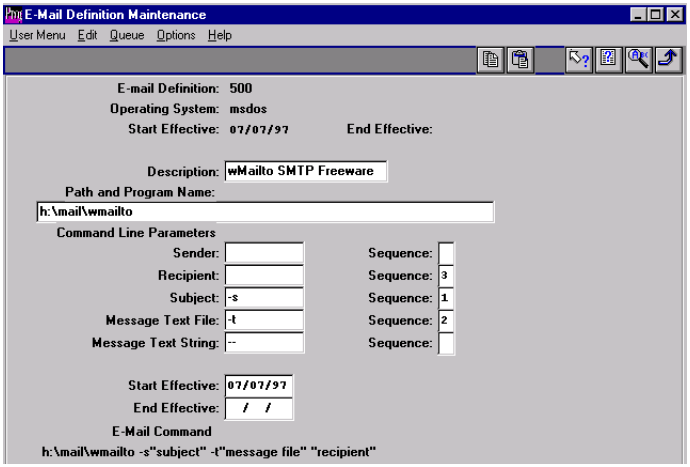


Fig. 4.10
E-Mail Definition Maintenance (36.4.20)

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 or Windows NT 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 eB 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 logon 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 “Defining Users and User Groups” on page 38.

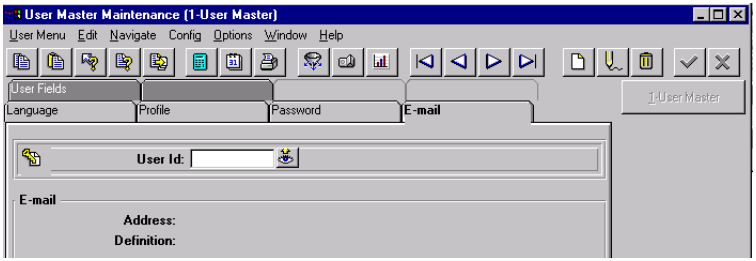


Fig. 4.11
User Maintenance (36.3.18)

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:

- Single database implementation, see *Implementation Process Guides: Main Tasks* and *Subtasks*.
- Consolidated order processing, see *User Guide Volume 2: Distribution*.
- Distribution requirements planning, see *User Guide Volume 5: Supply Chain Management*.

Each MFG/PRO eB database contains the following:

- A set of master files for customers, items, and sites
- A set of site-based inventory files
- 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 files 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.

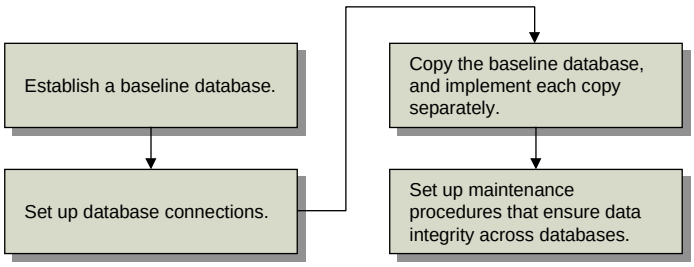


Fig. 5.1
Multiple Database Setup

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 eB. Load your data into this database.
2. Use Language Code Maintenance (36.4.1) to prepare the language master file for the language version you plan to use.
3. Set up data common to all databases to be implemented, as shown in Figure 5.2.

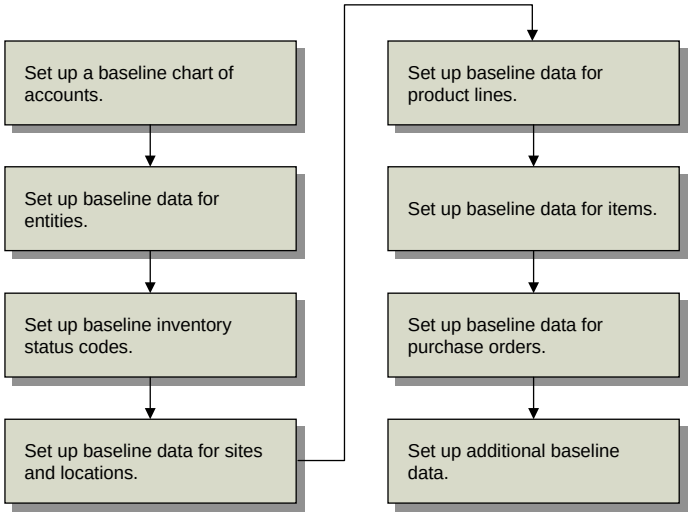


Fig. 5.2
Multiple Database Data Setup Work Flow

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 File
- Intercompany transactions
- Inventory
- Product lines
- Accounts Receivable
- Accounts Payable
- Purchase orders
- Sales orders. However, 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 an Intercompany Transfer account in the Accounts Payable Control File (28.24) and the Accounts Receivable Control File (27.24). Make sure this account is the default in the Inventory Control File (3.24). 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 the Inventory Control File.

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

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

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).

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 File.

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 files 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 the Inventory Control File. 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 the Purchasing Control File 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.

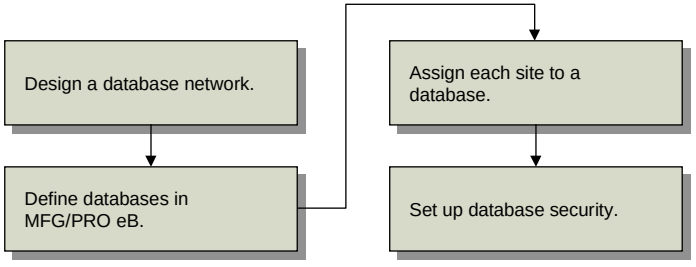


Fig. 5.3 Database Connection Flow

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.

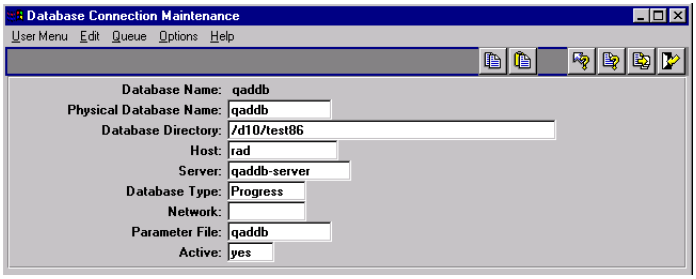
See "Managing Multiple Databases" on page 81.

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

Defining Databases in MFG/PRO eB

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

Fig. 5.4 Database Connection Maintenance (36.6.1)



Database Name. Enter the name of an MFG/PRO eB 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 start-up 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.

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.

Tip The pathname is case-sensitive.

Database Directory. Enter the full pathname 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.

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 NT, 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 81.

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 "Start-up and Shutdown" in the *PROGRESS System Administration Guide*.

Active. Enter Yes to have the primary database attempt connection with this database at start-up. 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 81 for details.

In operation, the security system on any database applies only to people logging into the database as their primary database. You can modify security for individual databases after you copy them.

Copying and Connecting Databases

1. Copy databases to the appropriate locations on your network using `prodb` or `mfgdb`. 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 82 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 the System/Account Control File (36.1). If needed, use Entity Code Maintenance (25.3.1) 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 2: 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 Type 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 the Purchasing Control File (5.24) and Sales Order Control File (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 the Security Control File (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 Start-Up Connections

Use Database Connection Maintenance (36.6.1) to set default connections for each database. If you do not want databases connected automatically at start-up, 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 files.

For each database involved in intersite requests, create sub-directories under the database directory using the names of the databases you connect to. Give them write permission. Intersite request records are written to these sub-directories 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 eB 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 into the primary database with a shell script, then use Current Database Change (36.6.17) to switch databases.
2. Log into each database using a different login script.

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

Example To log into Database2 located on the Database2 CPU, a login 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 eB.
-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 login 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 eB 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 changing databases, you retain the security assigned to you by the primary database (the first one logged into). For example, if you can execute inventory transactions in the primary database, you can do so in every connected database.

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 sub-directory, where you can access them later.

Controlling User Logins

Combinations of PROGRESS and MFG/PRO eB 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 login 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 login 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 login script. All sales orders filled in Database1.
Inventory Controllers	Database2 login script. Their work does not require access to other databases.
Master Schedulers	Database1 and/or Database2 login script, depending on security. Ability to access Current Database Change.
Shippers	Database1 login script. Shipments are recorded in Database1.
Purchasers	Either Database1 or Database2 login script, depending on site used.
Receivers	Database2 login script. Receipts are recorded in Database2.

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

System Security

At the database level, two kinds of security are possible: menu security and field security. Menu security controls a user's ability to execute programs associated with a given menu. Field security controls a user's ability to modify information in individual fields.

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

For each CPU, menu security is active once a user passes the login screen. MFG/PRO eB reads the password or login ID, and checks the primary database for security settings associated with that password. If a program is made unavailable to a given password or login ID, that restriction applies for the entire session, even when the current database is changed.

Field security is activated whenever a secured field is accessed. The system checks permissions based on user IDs. Unlike menu security, field security is determined by rules specified in the current database.

Table 5.3 illustrates some possible security configurations. In the table, the primary database is the one initially logged into. The user then changes to the current database using Current Database Change. The login name is the ID used to log into the current CPU.

Table 5.3 Multi-Database Security Options

Users	Login Method	Menu Security	Field Security
Single-database user	Single script.	Primary database.	Primary database.
Multiple-database user	Log into primary. Use Current Database Change.	Primary database.	Current database by login name.
	Log into each with different scripts. Current Database Change is disabled.	Varies with login script.	Current database by login name.

Users	Login Method	Menu Security	Field Security
	Log into each CPU first. Use scripts on that CPU.	Primary database, as specified by login script.	Current database by login name, which changes with change of CPU.
	Login script specifies number of databases connected to. Database Connection Maintenance is disabled. Database Connect (36.6.13) is accessible.	Primary database.	Current database by login name.

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 orders 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 files such as customers, salespersons, items, credit terms, freight lists, and price lists from the sales order database.

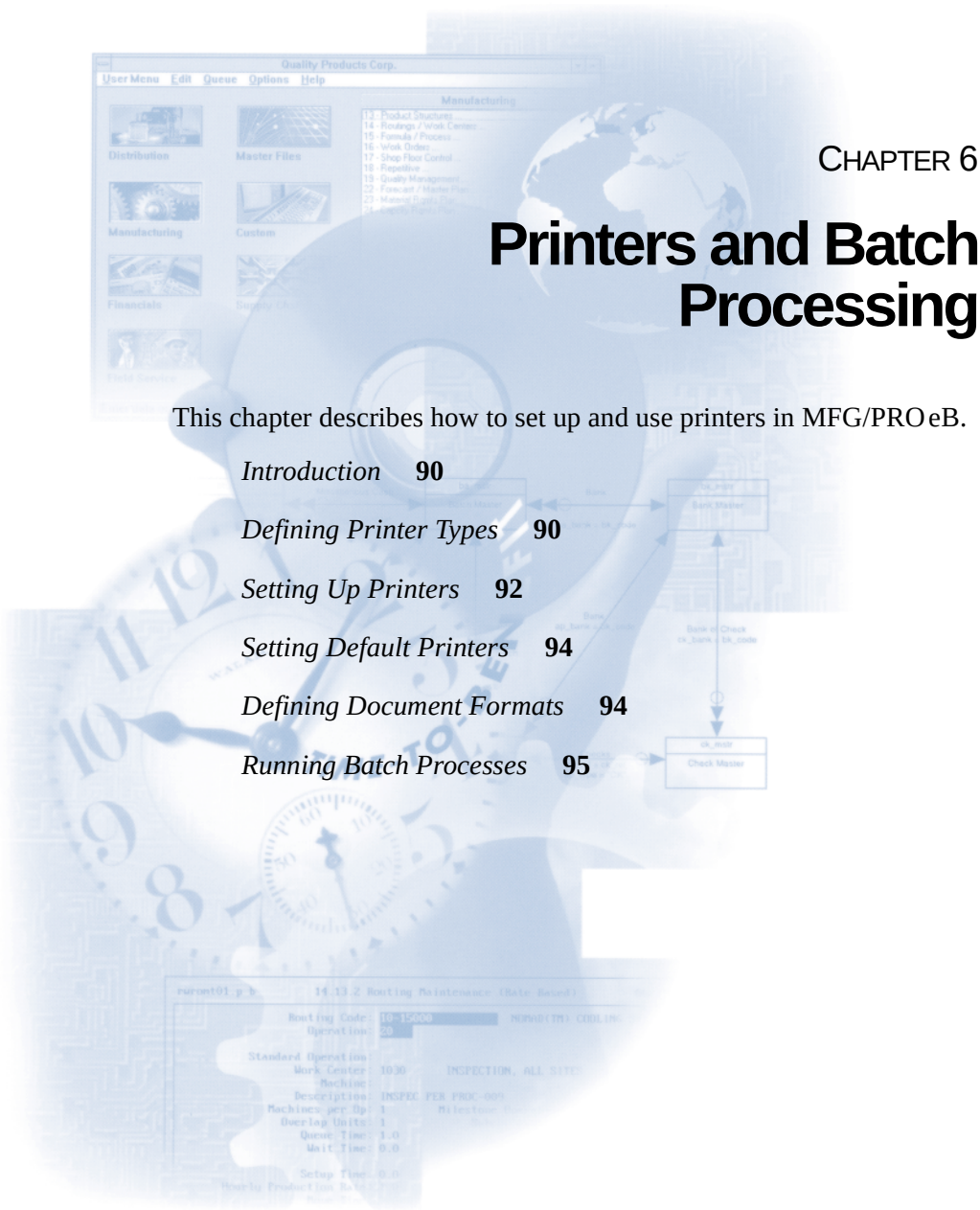
When the site belonging to the inventory database is specified, the system looks at the site master, determines which database the site belongs to, then checks the database connection master (dc_mstr) to see if that

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 files, 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, files such as customer master and credit terms in the sales order database are used, and accounts associated with those files 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 eB.

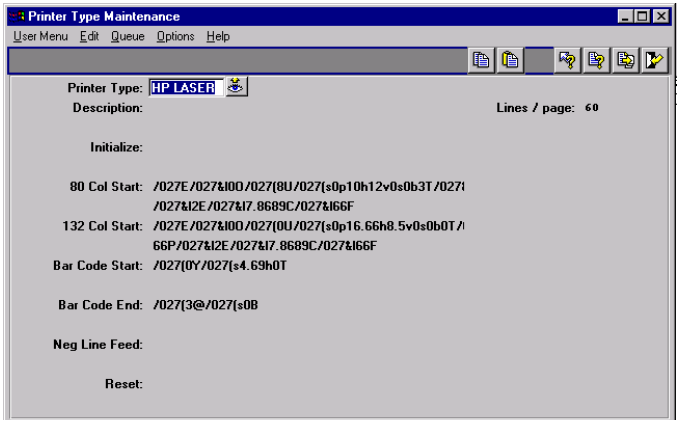
Introduction

You can send reports, files, 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. To schedule batch jobs at the operating system level, see the appropriate *System Administration Guide.*

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
- Bar-code 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

The demo database contains the correct control sequences for some commonly used printers. To use that data, dump the printer master (pr_mstr) file from the demo database and load it into your database.

Note One of the default printers is the terminal. Use terminal in a character interface and window in a Windows interface.

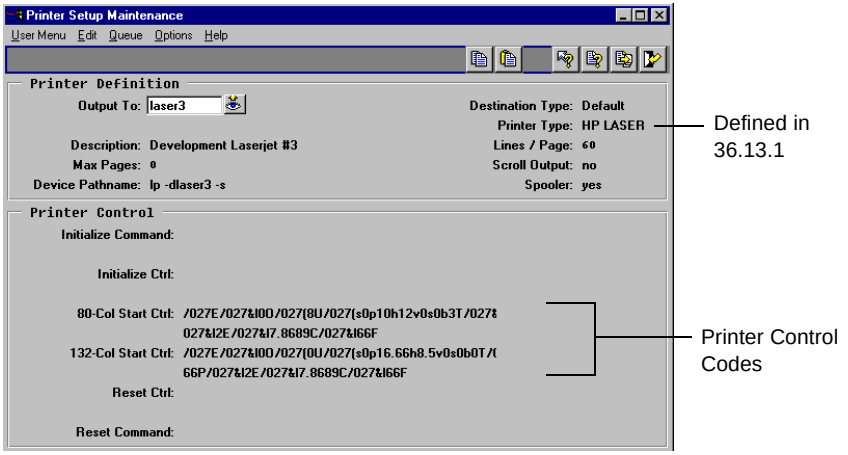
Table 6.2
Sample Printer Control Codes

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 eB 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, or a Windows display. In Language Detail Maintenance, 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 64.

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 pathname 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 pathname. Table 6.3 lists examples of device pathnames.

Table 6.3 Sample Device Pathnames

Device Pathname	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 pathname is:

```
/device/tty
```

Setting Default Printers

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

Use Printer Default Maintenance (36.13.4) to assign default output devices to users. 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, 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 eB 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 Output as a batch ID rather than a printer ID.

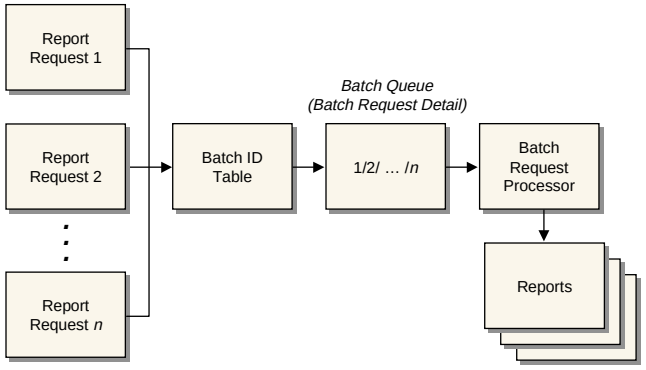


Fig. 6.3
Batch Processes

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 orders 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 eB. You can run the batch file (the `mpro` program) automatically with `cron`. For information on how to do this, refer to the appropriate *System Administration Guide*.

CHAPTER 7

CIM Interface

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

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

See page 115.

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

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: Runtime 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 bar-code readers.

If data is loaded directly into files using dump/load programs or PROGRESS loads, some files may not update correctly.

Load data into MFG/PRO eB using function on the CIM Interface Menu (36.15). Imported data can come from:

See “CIM Data Format” on page 100.

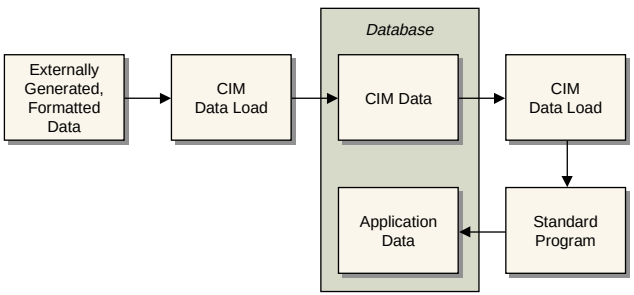
- Any ASCII file that follows the correct conventions.
- The output of programs that run in multiprocessing environments such as UNIX.

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 file. The CIM interface enables you to construct a file of input values for Product Structure Maintenance (13.5), 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 files. In OBCM applications, go to Options in the menu bar and select Alternate Input, then Direct CIM Load Input. 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 files through the appropriate input screen.

Fig. 7.1
CIM Data Load

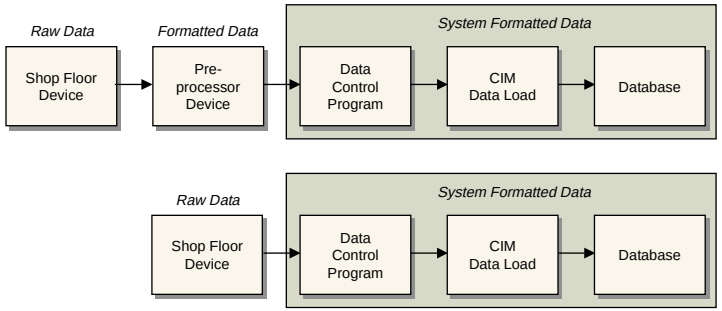


When CIM Data Load reads a data load group, it creates a record in the batch data load master file and assigns it a unique group ID. This integer record contains the name of the MFG/PRO eB program to receive the data, and the date and time when the record was added. CIM Data Load then creates a record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

See “CIM Data Format” on page 100 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

Each program takes in data in groups. A group consists of highlighted fields within a folder. Data going into the CIM load must use the rules described in this section.

See “CIM Data Input File Example” on page 103.

The `@@BATCHLOAD` key word signals the beginning of the data-load group, consisting of one or more lines. Program name is the MFG/PRO eB 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 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.
- Separate fields with spaces.
- Surround character fields with quotation marks.
- The end of an input line performs the same function as the Go key. Fields for which there are no data, and which come at the end of an input sequence, do not require hyphens.
- Type all characters in lowercase, taking care to spell correctly.
- Separate each field with a space. This executes a Tab command, which skips a field, retaining the default or existing value, and does not require quotes.
- Use a hyphen (-) to execute a Tab command. For example, to accept default data for fields 1, 2, 3, 5, and 7, and enter Yes, 12, and 01/01/99 for fields 4, 6, and 8, enter the following:

  ```
  --- “yes” - “12” - “01/01/99”
  ```

- Skip data at the end of an update group by ending the line, much like a Go keystroke.
- Format data as it is entered.
- Use a period on a line by itself to indicate End or End-Error.

  For repeating input (that is, multilevel), use the period to go back one level. This executes the End command.
- At the end of each input group, use a line feed. This executes the Go 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 start-up parameters (-d parameter).
- Use a caret (^) to indicate a null value.

Tip
F5 cannot be simulated.

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 same 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 four entry groups, corresponding to the number of windows. Although direct correspondence between entry groups and windows is normal, it is not required. The four entry groups are:

- Key field group—employee code
- Personal group
- Address group
- Employment 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.
```

CIM Format for Object Programs

Important CIM load processing is not designed to load records into the object programs on the Component Configurator menu (36.20.13). For example, you cannot use CIM load functions to copy a browse created in Browse Maintenance from one database to another.

For object programs, follow the procedure for importing data through the CIM Interface. You have the option of adding lines to the CIM load. The lines enclosed in square brackets are optional.

```
@@BATCHLOAD program-name
[@LOADPROC load-program]
[@ERRORDATAFILE error-out-file]
[@SUCCESSDATAFILE success-out-file]
[@DEFAULTSFILE defaults-file]
@ACTIONEDIT or @ACTIONDELETE
(input data)
@PROCESS
@@END
```

The `@LOADPROC` key word lets you use a PROGRESS procedure to filter data prior to sending it to program-name. All data between the `@LOADPROC` and `@PROCESS` lines are sent through the load-program procedure. This procedure can be an internal screen process or an external PROGRESS program. In either case, the procedure takes a single input-output character parameter that contains the current line for the CIM session.

If the requested load-program cannot be found, you are prompted to either continue the CIM load without external processing or cancel processing for this screen. Note that `@LOADPROC` is set only for the current screen. After an `@PROCESS` is encountered, a new `@LOADPROC` is required if additional user screen processing is required.

The `@ERRORDATAFILE` and `@SUCCESSDATAFILE` key words allow for generation of CIM data-load files for CIM load processes that either succeeded (success-out-file) or failed (error-out-file). If these files exist, they are overwritten with the results of the current CIM load process. Once the file name of `@ERRORDATAFILE` or `@SUCCESSDATAFILE` has been defined, it remains in effect until changed or until the end of the current CIM load process. To turn off either of these options, do not define a file name. Instead, assign a blank file.

The @DEFAULTSFILE key word lets you set up a file of default values that are applied to fields in the data-groups that follow the @DEFAULTSFILE key word that the CIM data load is processing.

Use the following format for the defaults file:

```
program-name
group-name field-name field-value
```

The program name indicates the program the defaults file applies to. This is the same program name that follows @BATCHLOAD. The defaults file is applied only to those groups that are processing data through the same program.

When combined with the program name, group-name and field-name identify the field to move the value in field-value to. The easiest way to generate a defaults file is to use the Alternate Output option under the Options menu and select CIM Flat File. This generates a file you can edit to create the desired defaults. The only entries in the defaults file should be those replacing field values in the CIM load.

Note Although you cannot use CIM load with the object programs on the Component Configurator menu (36.20.13), the Options menu is the same for all object programs. The system lets you select CIM Flat File in the Alternate Output window to generate a file containing all groups and fields for Component Configurator menu programs. However, the resulting CIM file will fail the load process.

The @DEFAULTSFILE key word is additive, meaning that its subsequent use within a CIM load file adds default files to any already in existence. To remove all default files from subsequent group ID loads, set the default file to a blank file (for example, @DEFAULTSFILE).

The @ACTION key word determines what is to be done with input data. @ACTION EDIT means the data modifies the record, and the database is be updated. @ACTION DELETE means the record is removed from the database.

Only data following the @ACTION key word and preceding the @PROCESS key word is processed. Data outside the two key words are not processed. Other key words, such as @DEFAULTSFILE, are processed outside the CIM process.

Each data element or field value in a group is identified by its program name, group name, and field name. The group name is the name of the group in the program function that contains the field name. The field value is the value assigned to this field. To obtain a list of group and field names for a particular screen, select Alternate Output under the Options menu. Select CIM Flat File in the Alternate Output window to generate a file containing all groups and fields for the screen.

The `@PROCESS` key word identifies the point where the data group is complete for the given screen.

The `@@END` key word tells the program to begin internal processing of data for validity, and generate a single PROGRESS transaction.

In the example below of a CIM data load to an object program, data is loaded through the program `cmcn001.w` (Cash Flow–Misc. Maintenance).

The `@ERRORDATAFILE` key word sends CIM data containing errors to `c:\mfgpro\adi\error.out`. If this file already exists, it is overwritten. Similarly, the `@SUCCESSDATAFILE` key word sends successfully loaded CIM data to `c:\mfgpro\adi\ok.out,` which is also overwritten if it exists. The `@DEFAULTSFILE` designates a file to use for overlaying values contained in this CIM flat file. In this case, `c:\mfgpro\adi\template\master.tmp` provides information that remains constant for this CIM data load.

The `@ACTION` key word denotes the start of screen data processing; in this case screen data is modified, rather than deleted.

The lines following the `@ACTION` key word are all in the form group-field-value. For example, in the first screen data line, the group is Main (as are all groups in this example), the field is `bk_mstr.bk_desc`, and the value assigned to the field is BKW Bank Account #8293-01. The `@PROCESS` key word tells the CIM processor that all screen data elements have been assigned, and that the screen can begin internal processing of data for validity.

```
@@BATCHLOAD cmcn001.w
@ERRORDATAFILE c:\mfgpro\adi\error.out
@SUCCESSDATAFILE c:\mfgpro\adi\ok.out
@DEFAULTSFILE c:\mfgpro\adi\template\master.tmp
@ACTION EDIT
"Main" "bk_mstr.bk_desc" "BKW Bank Account #8293-01"
"Main" "cf_mstr.cf_bank" "AA"
```

```
"Main" "cf_mstr.cf_curr" "USD"
"Main" "cf_mstr.cf_due_date" "02/07/00"
"Main" "cf_mstr.cf_entity" "1000"
"Main" "cf_mstr.cf_ent_ex" "1.00000"
"Main" "cf_mstr.cf_expt_amt" "0.00"
"Main" "cf_mstr.cf_expt_date" "12/24/00"
"Main" "cf_mstr.cf_lastedit" "02/04/93"
"Main" "cf_mstr.cf_nbr" "3"
"Main" "cf_mstr.cf_ref" "bank interst-curr"
"Main" "cf_mstr.cf_type" "yes"
"Main" "cf_mstr.cf_user1" "QQQ"
"Main" "cf_mstr.cf_user2" "QQQ"
"Main" "cf_mstr.cf_userid" ""
"Main" "cf_mstr.cf__dec01" "0.00"
"Main" "cf_mstr.cf__dec02" "0.00"
@PROCESS
@@end
```

A simple example of `@LOADPROC`, a user-defined PROGRESS program for processing alternate input, is shown below. This program takes each string representing a group-field-value identifier for a screen element and breaks it into each of its components to be displayed through a message box. Group, field, and value fields are separated by spaces, and surrounded by double quotes.

```
/*********************************
PURPOSE: Sample program for providing user-
defined processing of re-directed CIM output.
PARAMETERS:
INPUT-OUTPUT c-data contains "group-name" "field-name" "field-
value" as a character string.
NOTES:
Each visual screen element will be sent through this procedure in
the preceding group-field-value format described by c-data. In
this simple example, the string is broken into its components and
displayed through a message box.
/*****************************************
define input-output parameter c-data as character no-undo.
define variable c-group as character no-undo.
define variable c-field as character no-undo.
define variable c-value as character no-undo.
define variable c-temp as character no-undo.
/* Extract group, field, and value from c-data string */
if index(c-data," ") > 0 then
    c-group = substring(c-data,1,
    index(c-data,"     ")).
else
    c-group = substring(c-data,1,length(c-data)).
    c-temp  = substring(c-data,length(c-group) +
    1,length(c-data)).
if index(c-temp," ") > 0 then
    c-field = substring(c-temp,1,
    index(c-temp," ")).
else
    c-field = substring(c-temp,1,length(c-temp)).
```

```
    c-temp  = substring(c-temp,length(c-field) +
    1,length(c-temp)).
/* If only two fields, assume field and value (Group is optional)
*/
if c-temp <> "" then
    c-value = substring(c-temp,1,length(c-temp)).
else
    assign
        c-value = c-field
        c-field = c-group
        c-group = "".
/* Get rid of white spaces and outermost quotes */
c-group = trim(trim(c-group),'"').
c-field = trim(trim(c-field),'"').
c-value = trim(trim(c-value),'"').
message
    "Group:  " c-group skip(1)
    "Field:  " c-field skip(1)
    "Value:  " c-value
    view-as alert-box.
return "".
```

Creating a CIM Input File

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

1. Run the program that is to receive the data.

   a. For non-object programs, note the name of the program, shown in the upper left corner of the screen.

   b. For object programs, either run Menu System Report (36.4.5) or open the Help About screen to determine the name of the opening program.

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. Perform a lookup 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.

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

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).

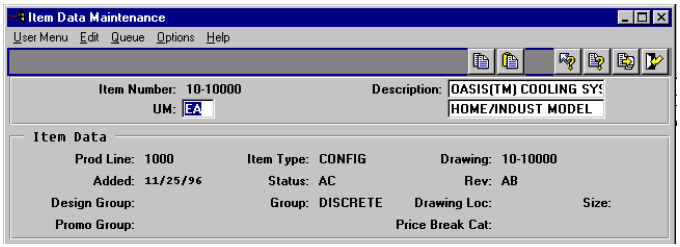


Fig. 7.3
CIM File for Item Data Maintenance (1.4.3)

Use the following code to load this data.

```
@@batchload ppptmt04.p
“10-10000”
“Ea” “Oasis(TM) Cooling System” “Home/Indust Model”
“1000” “11/25/96” “HVAC” “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 any errors.

- 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?

Exporting Data Through Alternate Output

When exporting data, use the Alternate Output Options dialog box to specify a destination for output data, how much data to send to that location, and whether to process the output data interactively. You can filter exported data through a user-defined PROGRESS procedure.

Tip
The destination should be a file or PROGRESS program.

Figure 7.4 illustrates the export process.

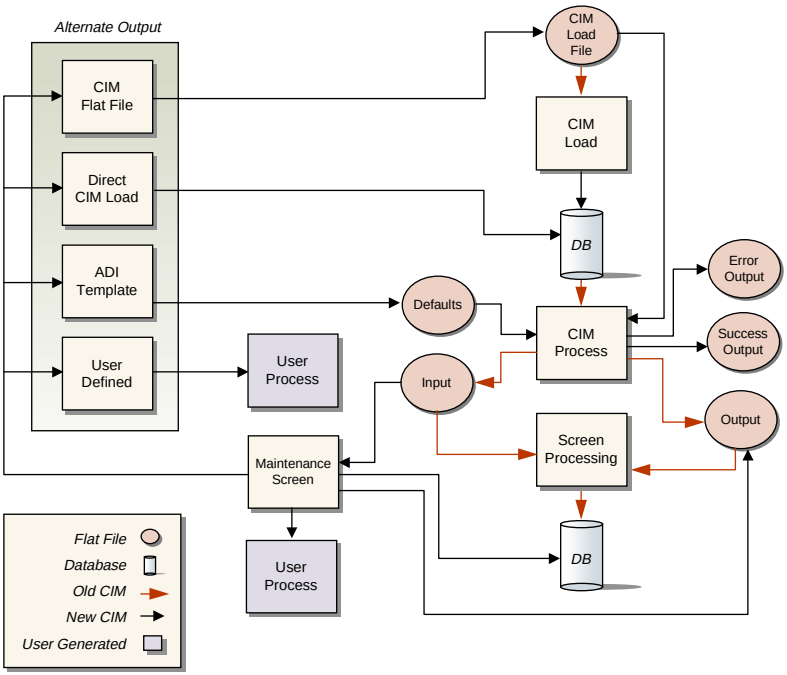


Fig. 7.4
Alternate Output Options Flow

Alternate Output Options Window

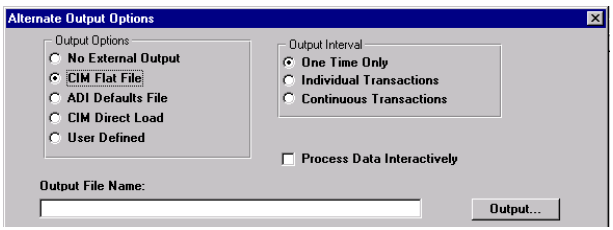


Fig. 7.5
Alternate Output Options Window

Tip
ADI stands for Application Data Interface.

No External Output. Choose this to turn off alternative output. The CIM Flat File option creates a basic new CIM format file that can be sent through CIM data-load processing (36.15.1 and 36.15.2).

ADI Defaults File. Choose this to generate a defaults file to use with the `@DEFAULTSFILE` key word for CIM data loads.

CIM Direct Data Load. Choose this to load screen data directly to CIM database tables. This is the same as generating the CIM flat file and loading the resulting file through 36.15.1, but does not generate the intermediate flat file.

A sample procedure is shown on page 113.

User Defined. Choose this to send screen data to another program. The program must be either an internal procedure for the targeted window or a PROGRESS procedure. For internal procedures, parameters are:

- Name of output file
- Name of executing window
- Name of group in the group
- Current value of field

One Time Only. The first transaction (one screenful) is processed to the alternate output, and subsequent transactions are processed with no external output.

Individual Transactions. External output is processed and generated one transaction at a time.

Continuous Transactions. External output is generated after processing of all individual transactions is complete. Use this for interdependent data.

Process Data Interactively. Choose this to process data and update the database before data are sent to the alternate output.

Sample User-Defined Output

The following simple example shows a user-defined PROGRESS program for processing alternate output. This program passes five arguments for each field in the application: user-selected output file name, name of application, name of group to which the field belongs, name of field, and value currently assigned to the field.

```
/*************************************
PURPOSE: Sample program for processing alternate output through
a user-defined program.
PARAMETERS:
 INPUT c-ext-file name of output file provided by user
INPUT c-file name of object application (e.g., adcn001.w
INPUT c-group group name for this field
INPUT c-field name of field
INPUT c-value value for this field
NOTES:
Each data element is sent through this procedure with its
associated group and value, as well as the application file name
and name of the output file. This example displays the output
file, application file, group name, field name, and field value
in a message box.
*****************************************/
define input parameter c-ext-file as character no-undo.
define input parameter c-file as character no-undo.
define input parameter c-group as character no-undo.
define input parameter c-field as character no-undo.
define input parameter c-value as character no-undo.
message c-ext-file c-file c-group c-field
c-value.
```

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 start-up 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. MFG/PRO eB is not notified when the operating system kills a session, and a record of the session may still be shown by the CIM Process Monitor.

The best way to kill a CIM session is to use the CIM Data Load Process Monitor (36.15.4). 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 the F5.

CHAPTER 8

Database Management

MFG/PRO eB provides utilities for monitoring database size, performing dumps and loads, reloading archive files, managing database sequences, and monitoring licensing and user count.

Managing Database Size

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

Determining Disk Usage

Tip
The program requires adequate free disk space to run.

Use Database File Size Inquiry (36.16.1) to dump selected files and review their file sizes. Reported file 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 NT 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 files in a database are history, sales order, and purchase order files. 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 119.

Dumping and Loading Data

Dump/load programs move the contents of data files into or out of ASCII files. The dump procedure reads a database file, 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 file (*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

See “Using the CIM Interface” on page 98 for details.

flow and format the CIM input is duplicating. The *Database Definitions* book contains details on specific file formats.

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 MFG/PRO eB data.

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 116.

3. Log into MFG/PRO eB in single-user mode. You can speed up the dump/load by running multiple sessions of Database Dump/Load from multiple terminals.
4. Execute Database File Dump/Load for the correct range of files.

   If there is enough free space, select all files. 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 eB database (mfg) onto your old database.
6. Load the dumped files back into the database using Database File 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 program is run in a window similar to a report criteria input screen. You choose records based on selection criteria you specify. 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	Installed Base History	RMA History
Accounts Receivable	Intersite Requests	Routings
Audit Detail	Intrastat History	Sales Analysis
Closed Cumulative Orders	Invoice History	Sales Order Shippers
Closed Intersite Demand	Lot Masters	Self Bills
Closed PO Shippers	Master Bills of Lading	Service Contracts
Closed Projects	NRM Sequences	Service/Repair Orders
Closed Purchase Requisitions	Operation History	Shippers
Closed Purchase Orders	Operation Plans	Subcontract Shippers
Closed Purchase Receipts	Physical Inventory Tags	Supplier Performance Data
Closed Service Calls	Product Change Orders	Supplier Schedules
Closed Service Requests	Product Change Requests	Transaction History
Customer Schedules	Product Structures	Uninvoiced Receipts
Expired Sales Quotes	Quality Orders	Window Help
Expired Call Quotes	Quality Test Results	WIP Lots
Family Hierarchies	Repetitive History	Work Orders
Forecast Details	Retired Fixed Assets	Zero Inventory Balances
GL Transactions		

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, file, and field, it keeps the latest record and deletes/archives the rest.

To delete and/or archive files:

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 on-line or in the .hst file. If records selected for deletion are correct, delete the .hst file, and 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 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.

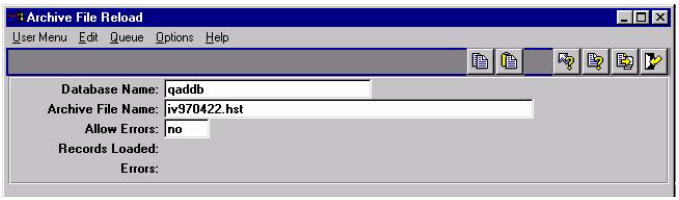


Fig. 8.2
Archive File Reload (36.16.5)

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.

Managing Database Sequences

When a unique identifier is needed by an MFG/PRO eB program, the system often uses a control file 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 file 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 File 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 files 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.

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

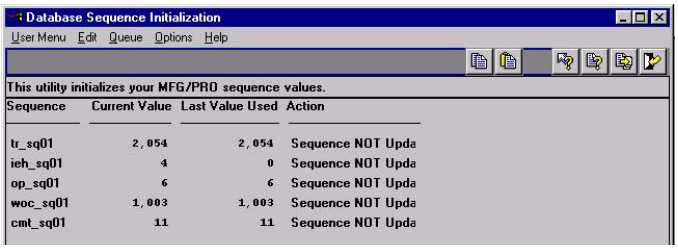
To guarantee database integrity, perform sequence maintenance:

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

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 127.

Maintain sequences in Sequence Maintenance (36.16.13). Sequence Maintenance works with PROGRESS RDBMS only. Oracle dataservers are not currently supported.

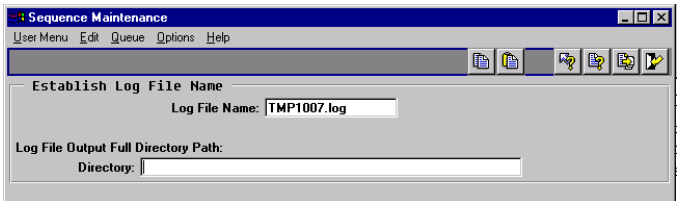


Fig. 8.4 Sequence Maintenance (36.16.13), Establish Log File Name Frame

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

Directory. The OS directory where you want to store the file.

A second Sequence Maintenance screen appears.

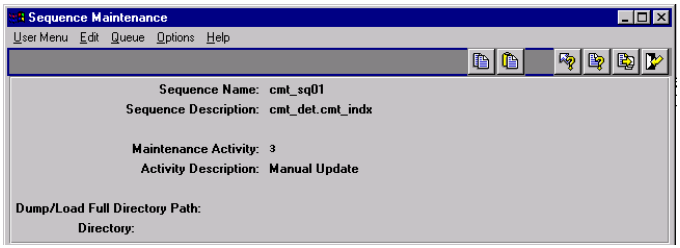


Fig. 8.5 Sequence Maintenance, Sequence Name Frame

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.

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.

- 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.

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

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.

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

Maintaining Sequences Using CIM

For more information on CIM, see "Using the CIM Interface" on page 98.

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 eB function. This value is not available for processing, since it was consumed by another process.

Values used to update a sequence are validated against a range of acceptable values for the sequence, as established by MFG/PRO eB. 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 eB 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 eB 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.

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 eB function.

See“Maintaining Sequences Manually” on page 123.

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;
```

Monitoring Licenses and User Count

MFG/PRO eB provides utilities that enable the system administrator to monitor the number of logged-in users. These utilities also alert you to license violations and to software expiration. License utilities do not currently restrict the number of users that can log in.

Note Licensing applies only to modules licensed by user count. For example, it does not apply to the EDI module.

User Licensing System

When you receive MFG/PRO eB or an upgrade, you also receive a license code for each site. This code identifies the number of users for which the site is licensed. The system requests licensing information at three points:

- After installation, the system administrator must log into MFG/PRO eB and enter the license code.
- When a user logs in and the number of logged-in users exceeds licensed users, two error messages display. The user must press OK or Enter to exit each message.
- When a user logs in within 10 days prior to the software expiration date, a warning displays. After expiration, no users can log in and you must contact your distributor or QAD for a renewal or a temporary license code.

Named and Concurrent User Licensing

User counts are tracked by one of two licensing schemes: concurrent users and named users.

- In concurrent user licensing, each concurrent login is counted as a user. If a single user logs into multiple sessions simultaneously, each login is counted.
- In named user licensing, each physical user logging in is counted as one user, regardless of how many simultaneous sessions they have.

Table 8.2 is an example of QAD licensing schemes.

Table 8.2
Licensing Schemes

Physical Users	Named User License Count	Concurrent User License Count
John logs into a session.	Counts as one user.	Counts as one user.
Mary logs into a session.	Counts as one user.	Counts as one user.
Mary logs into another session.	Does not count as an additional user.	Counts as one user.
Total Physical Users: 2	**Total User Count: 2**	**Total User Count: 3**

User Licensing Programs

User licensing programs are provided on the Database Management Menu (36.16).

License Registration

Use License Registration (36.16.10) at initial login to register your new license code or at any other time to edit an existing license. When your software expires, access to MFG/PRO eB is limited to this program.

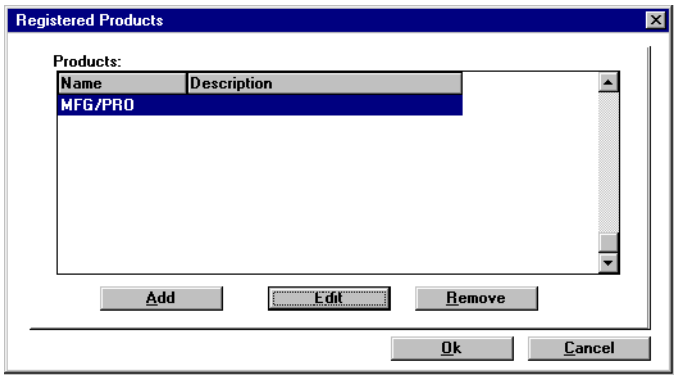


Fig. 8.7
License Registration (36.16.10)

License Violation Report

The License Violation Report (36.16.11) displays a list of users who have logged into the system in excess of the allowed user count. The date and time of the login display.

User Monitor Inquiry

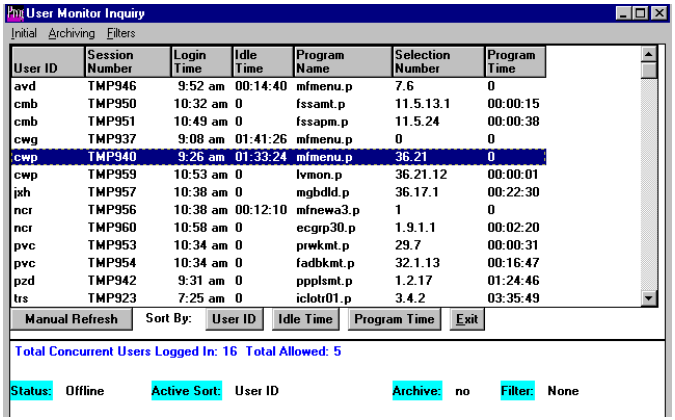
User Monitor Inquiry (36.16.12) displays users currently logged in along with:

- Date and time of login
- The program they are currently executing
- Program execution time
- Amount of time they have been idle if no program is selected

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

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

Fig. 8.8
User Monitor Inquiry (36.16.12)



Initial Menu

Update inquiry information manually using the Manual Refresh button or at an automatic refresh rate using the Initial pull-down menu.

To set automatic refresh rates:

1. Choose Set Auto Refresh Rate from the Initial menu.
2. Enter the refresh rate interval in the Set Refresh Rate In Seconds frame, and choose OK.
3. Enable the auto refresh feature by choosing Start Auto Refresh on the Initial menu.

Note When auto refresh is active, you cannot close the User Monitor Inquiry. The Archiving and Filters menus are also disabled. Use the Stop Auto Refresh button to return to manual refresh.

Archiving Menu

Use the Archiving pull-down menu to save a unique copy of inquiry information to a text file.

To start archiving, select the Archiving pull-down menu and choose Set Archive Parameters. In Archive Directory, designate the archive's location. Set the archiving interval in the How Many Refreshes Before Archiving field. For example, if you enter 2, an archive entry is created after every two refreshes. Archives are created only when auto refresh is active.

Filters Menu

Use the Filters menu to create and save multiple filters to perform searches based on specific criteria. Only one filter is applicable at a time; however, several criteria can be combined in a single filter.

To create filters, select the Filters pull-down menu and choose Maintain/ Create Filters. Enter filter criteria in the appropriate fields, and choose Create.

Use the Bind field when applying multiple filter criteria. To search for users who meet all the criteria, use And. To search for users who meet any one of the criteria, use Or.

Table 8.3 describes the User Monitor Inquiry's filter fields.

Table 8.3 User Monitor Inquiry Filter Field Descriptions

Filter Field	Description
From User ID	Select users by ID.
From Login Time	The unit of measure of this field is seconds from midnight. For example, to view users who logged on between 9 and 10 AM, enter 32400 (60 seconds x 60 minutes x 9 hours) in the From Login Time field and 36000 (60 seconds x 60 minutes x 10 hours) in the To field.
Idle Time in Menu for > =	Select users by the length of time (in seconds) users have been at a menu.
Time in Program > =	Select users by the length of time (in seconds) users have been at a program.

Filter Field	Description
Menu Selection	Select users by menu location.
Number of Sessions	Select users by number of sessions.

Sorting

Use the Sort By buttons to arrange information in the User Monitor Inquiry. To sort, choose the desired sort button: User ID, Idle Time, or Program Time.

Status Fields

The status fields indicate the current configuration of the browse features:

- Status indicates if the auto refresh feature is active (online) or inactive (offline).
- Active Sort indicates the sort currently in use.
- Archive indicates if the archiving feature is active.
- Filter indicates the filter currently in use.

CHAPTER 9

Reports and Utilities

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

Generating Master File Reports

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

Auditing Reports

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

See *User Guide Volume 4A: Financials.*

To maintain detailed information for a critical subset of master files, set Audit Trail to Yes in the System/Account Control File (36.1). Additional details are tracked for the following database tables:

- ac_mstr (Accounts)
- ad_mstr (Addresses)
- bk_mstr (Banks)
- cm_mstr (Customers)
- cu_mstr (Currency)
- flpw_mstr (Field Security)
- mnd_det (Menu Security)
- mu_mstr (Monetary Union)
- tx2_mstr (GTM tax rates)
- usr_mstr (Users)
- vd_mstr (Suppliers)

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

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

- Use Master File Audit Report (36.17.1) to print changed records in master files. The report includes the database file name, current version of the changed record, user ID of the person who made the change, and date.

- Use Master File Audit Detail Report (36.17.2) to show details about audited changes when Audit Trail is Yes in the System/Account Control File. The report includes current and previous versions of the record, with the time and date of any changes.

Note Auditing information for unposted general ledger (GL) transactions is maintained when GL Transaction Audit Trail is Yes in the General Ledger Control File (25.24). This data also displays on the Master File Audit Detail Report.

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 Files Report (36.17.6) to generate a report listing the current values defined for all control files 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 eB delete/archive functions. It does not delete each record specified. Instead, for each unique combination of user ID, file, and field, it keeps the latest record and deletes/archives the rest.

See “Audit Detail Delete/Archive” on page 120 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 file 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:

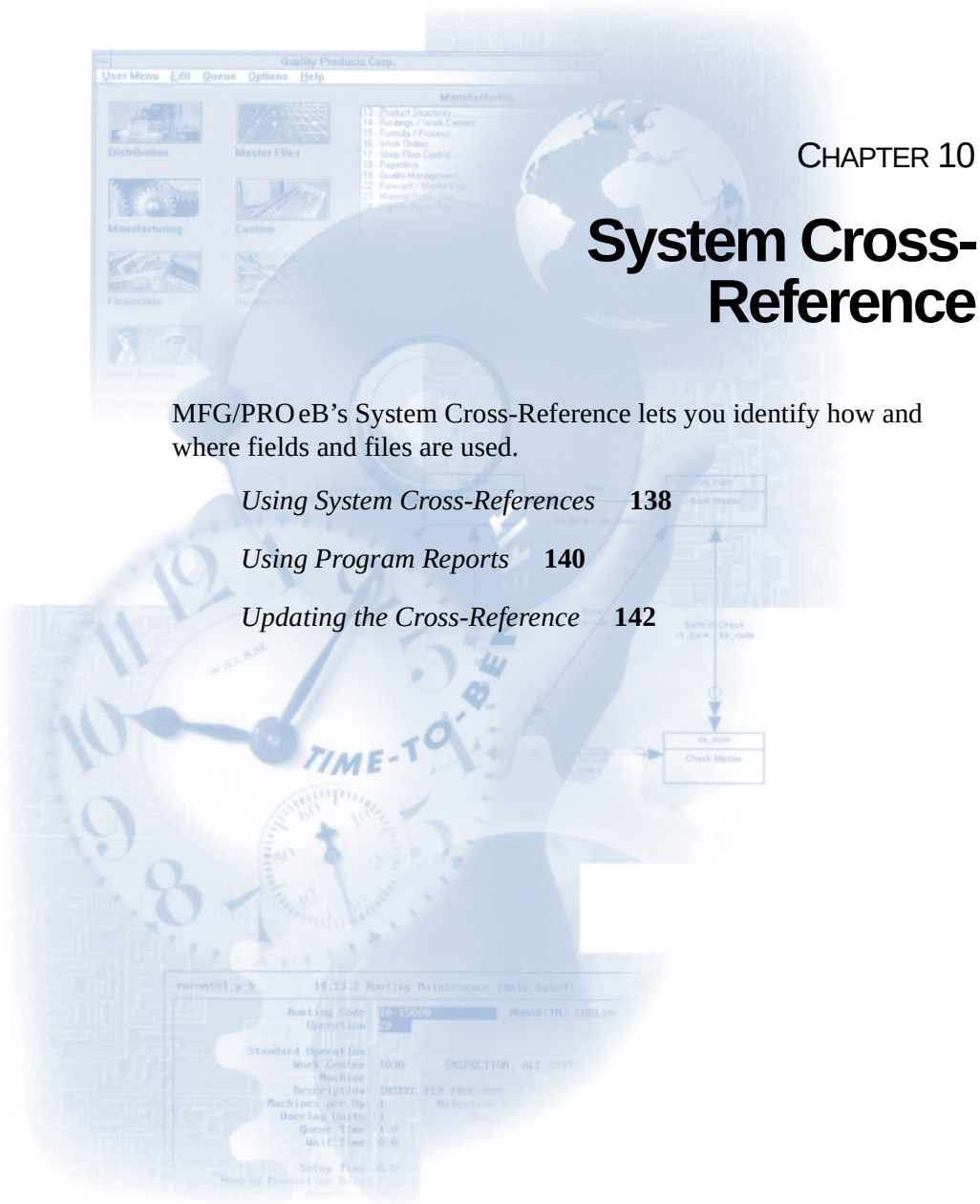
1. You are not using the General Ledger module to delete GL transactions created in other modules.
2. You implemented other modules prior to implementing the General Ledger. Before implementing General Ledger, delete the GL transactions in the unposted transaction file. These are reflected in the beginning 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 eB. 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 eB, 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 eB's System Cross-Reference lets you identify how and where fields and files are used.

Using System Cross-References

The System Cross-Reference menu (36.18) contains programs that identify how and where fields and files 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 eB, 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 files, fields, menu items, or programs. *Used* can mean referenced, updated, or called.
- Which files are referenced or updated by this menu item?
- Which menu items call this field?
- Which 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 eB 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 files, such as the item master (pt_mstr) or the printer master (pr_mstr). The database files consist of records containing entries in a group of fields.

When PROGRESS is compiled, the list of programs called and the files and fields read or potentially updated by those programs can be output. This output, along with MFG/PRO eB-supplied utility information, is the source of the cross-reference.

File, Field, and Menu Reports

The eight cross-reference reports answer such questions as “What does this file, field, message, menu item, or program do?” The syntax is XYZ. For example, the Files/Fields by Menu Report tells you what files 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 File, Field, and Menu Reports

Program Name	Description
Files/Fields by Menu Report (36.18.1)	Shows what files or fields are referenced or updated by programs called by a top-level menu. Limit searches further by execution file, database file, 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.
Files/Fields by Program Report (36.18.2)	Similar to 36.18.1, but not limited to menu-level programs. Shows what files 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 Files/Fields by Program Report for each relevant subprogram.
Menu Items by Field Report (36.18.4)	Shows which menu items call a field or range of fields. Further limit searches by execution file and database file. Shows field name and file, calling menu item, and kind of action performed. Action types are create, search, update, delete, and access.

Program Name	Description
Menu Items by File Report (36.18.5)	Similar to 36.18.4, but limited to a database file or range of files, rather than fields. Shows which menu items call a file or range of files. Further limit searches by execution file and menu item. Shows file 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 file name and program name. The report includes the following: • The name of the database file 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.

Program Name	Description
Programs by File Report (36.18.14)	Similar to 36.18.13. Shows all programs that call a particular database file or range of files. Further limit searches by program name. Useful when a file has changed.
Program Source File Report (36.18.16)	Creates a list of program components, or bill of material, for a specified program or range of programs. Shows all component parts, including nested executable files and include (.i) files, that are directly called by the specified programs.
Program Run Report (36.18.17)	Creates a multilevel list of components, or bill of material, for a specified program or range of programs. Shows all component parts, including nested executable files and include (.i) files, that are either: • Directly called by the specified parent program • Indirectly called by subprograms or include files that are, in turn, called by the specified parent program Use the Levels field to specify the number of levels to include in the report. For example, set Levels to 1 to list only the subprograms and include files directly called from the parent program.
Source File Where-Used Summary (36.18.19)	Shows which executable files use a specified source (.p or .w) 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 or .w) 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 Name	Description
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

User Interface Management

This chapter discusses programs that let you modify the ways users interact with MFG/PRO eB 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. You can also set up individual UI preferences for users, such as the type of menu that displays on Windows computers or the programs that are accessible from the User Menu.

Table 11.1 lists the user interface manager functions that are described in this chapter.

Table 11.1
UI: Manager Functions Programs

Number	Menu Label	Program
36.20.1	Drill Down Maintenance	mgcn001.w
36.20.2	User Tool Maintenance	mgcn002.w
36.20.3	Menu Substitution Maintenance	mgcn003.w
36.20.4	User Interface Profile	mgcn016.w
36.20.5	Window Config and Security Maint	mgcn019.w
36.20.13.10	Browse Maintenance	mgcn011.w
36.20.13.12	View Maintenance	mgcn008.w

This menu also contains programs that are not described in this chapter:

- If you are using the Network User Interface (NetUI) for the Java platform, additional programs support customizing the NetUI interface (36.20.10).

See the *Network User Interface Guide* for details.

- Several programs on the Component Configurator Menu (36.20.13) let developers create their own object-based programs. These programs follow the Object-Based Component Model (OBCM), used for some MFG/PRO eB maintenance programs.

See *User Guide Volume 1: Introduction*.

Maintaining Drill Downs

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 Maintenance (36.20.1) to assign drill downs to fields that do not have a drill-down browse, to replace a browse, or to delete one.

Before you can use this program, 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

To determine the name of the field and program, follow these steps:

See “Creating Browses” on page 162, for details on creating browses.

1. Run the program.
2. Select About… from the Help menu. The dialog box gives you the program name.
3. Return to the program and click on the field where you want to attach the browse. Press Ctrl+F and note the field name.

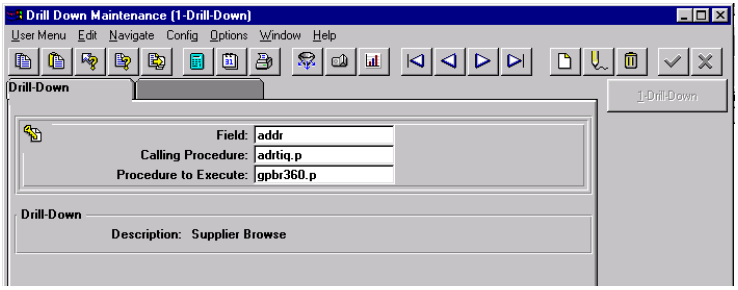


Fig. 11.1 Drill Down Maintenance (36.20.1)

You can assign more than one drill down to the same field. A menu of drill downs appears when you request the drill down.

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 Maintenance to associate a drill down with a field or program:

1. Enter a field name to associate with the drill down in Field. Leave it blank to associate it with all fields.
2. Enter the program containing the field in Calling Procedure.
3. Enter the browse name in Procedure to Execute.
4. Enter a description. Use a hyphen (-) as a qualifier. Do not use commas.
5. Choose Run to save.

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 tool bar and click on the field.
- Double-click on the field in the browse.
- Select the field and press Alt+F1.

Wildcards in Drill Down 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 Maintenance:

Field	`ad_addr`	`ad_addr`	`ad_addr`
Calling Procedure	`*`	`so*`	`soivmt.p`
Procedure to Execute	`adbrad.p`	`adbrcs.p`	`arbrbl.p`

When you drill down on `ad_addr` in `soivmt.p`, a menu shows all three browses: `adbrad.p`, `adbrcs.p`, `arbrbl.p`. When you drill down on `ad_addr` in a program other than `soivmt.p` but beginning with the

letters so, a menu shows two browses: adbrad.p and adbrcs.p. When you drill down on ad_addr anywhere else, the browse adbrad.p opens.

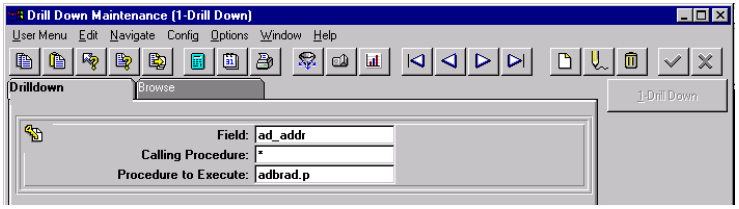


Fig. 11.2
Wildcards in Drill Down Maintenance

Drilling Down on Drill Downs

You can nest drill downs. In other words, one drill down can call another, which can call another, and so on. After creating the first drill down, you can assign the others to the same field. Enter the name of the first drill down as the calling procedure for the nested drill down.

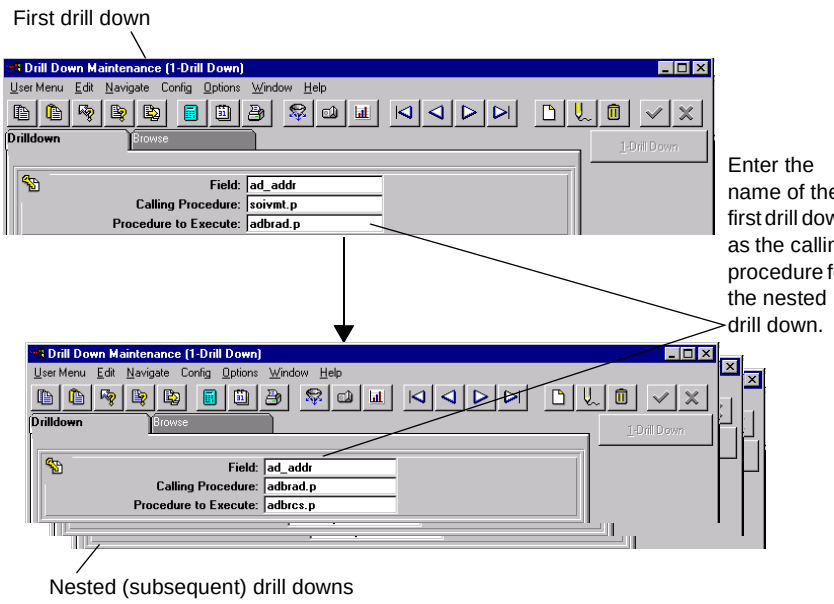


Fig. 11.3
Nested Drill Downs

Assigning Buttons to the Tool Bar and User Menu

With User Tool Maintenance (36.20.2), you can create GUI tool bar buttons to run programs. This makes it easier for you to run frequently used programs.

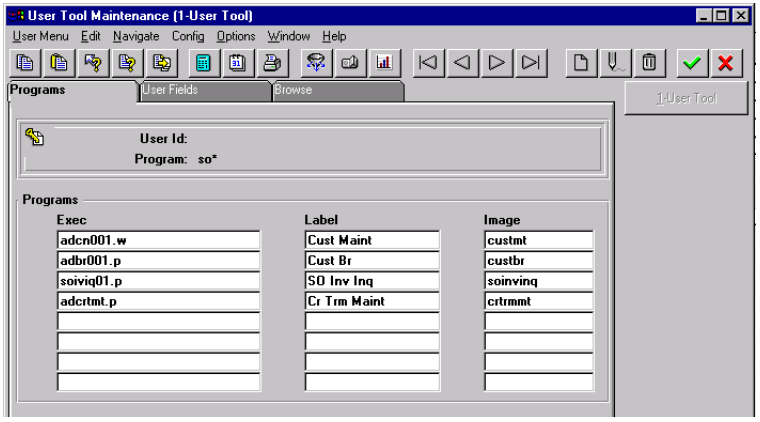
Note Tool bar buttons are not available from character terminals.

The Windows user interface is delivered with buttons on the tool bar in each program. With most programs, you can assign up to four buttons and four User Menu items to launch programs of your choice. In OBCM programs, you can assign up to eight buttons and eight User Menu items. The buttons are included in the tool bar on standard programs and on a separate bar in OBCM programs. Any standard and configured browses are suitable.

You assign programs to buttons 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.

Note From character terminals, the User Menu is accessible only in OBCM programs.

Fig. 11.4 User Tool Maintenance (36.20.2), Programs Folder



1 Enter a user ID or leave the field blank to assign to all users.

2 Enter a program name or leave the field blank to assign to all programs.

3 In the Exec fields, enter the program names (for example, `adcn001`) for the buttons to launch.

4 In the Label fields, enter the button labels, which you can write as abbreviated program names (for example, Cust Maint).

5 In the Image fields, enter the bitmap image file names. The image file must be in the user's PROPATH.

6 Choose the User Fields folder.

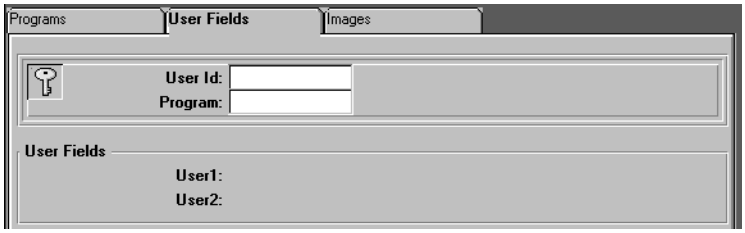


Fig. 11.5 User Tool Maintenance, User Fields Folder

7 User fields are customizable fields. System logic does not refer to these fields.

8 Choose  to save.

Setting Up Menu Substitutions

Use Menu Substitution Maintenance (36.20.3) 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.

Each user can turn menu substitution on or off in the Options menu or from User Interface Profile (36.20.4).

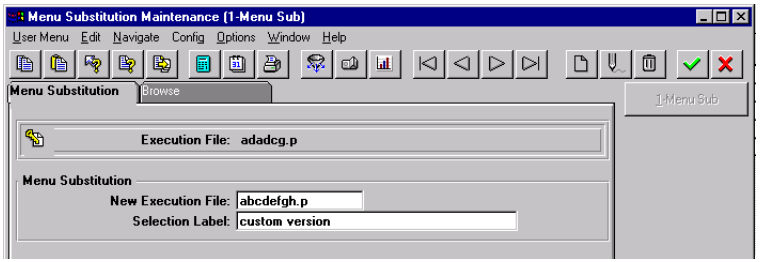
Menu substitution also affects standard MFG/PRO eB programs in these ways:

- Replaces browses with inquiry programs
- Replaces OBCM maintenance programs with the non-OBCM version, if it exists

Note If you are using the NetUI, menu substitution has the additional effect of replacing telnet programs with full Java-based GUI programs.

See the *Network User Interface Guide*.

Fig. 11.6
Menu Substitution Maintenance (36.20.3)



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 the label for the substitute program to display in Selection Label.

4 Choose  to save the record.

Setting Up User Interface Profiles

Tip
Most options apply to Windows environments. Only Auto Go and Menu Substitutions apply to the character interface.

Use User Interface Profile (36.20.4) to define default values for each user.

In Userid, select a user or group of users, or leave blank for all users. For that user or users, select:

- System tool bar position
- User tool bar position and whether to display it

See page 152.

- Menu style
- Background window color for OBCM programs. The system relates the number entered in this field to a color specification in the user's PROGRESS initialization file.

See page 151.

- Auto Go setting

- Appearance of online help. 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.
- Whether menu substitution is enabled or disabled

See “Adding User Help” on page 62.

Choose  to save your changes.

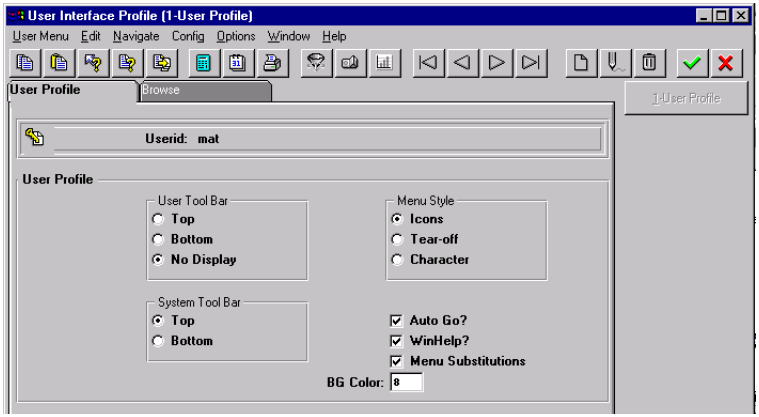


Fig. 11.7
User Interface Profile (36.20.4)

Selecting Auto Go

Select Auto Go? to change the actions of some navigation keys in OBCM programs (see Table 11.2 and Table 11.3).

Table 11.2
Auto Go Selected

	F2 (Windows) **F1** (Character)	**Enter**	**F12**
Key folder	edit	edit	edit
Data folder	next folder	next folder	commit
Last folder	commit	commit	commit

Table 11.3
Auto Go Not Selected

	F2 (Windows) **F1** (Character)	**Enter**	**F12**
Key folder	next folder	—	edit
Data folder	next folder	next folder	commit
Last folder	next folder	next folder	commit

Selecting a Menu Style

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

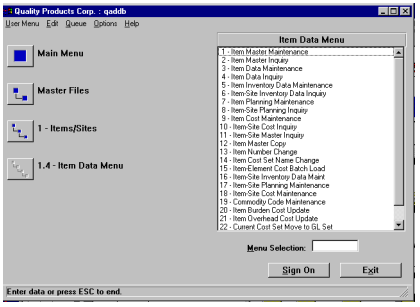
The Windows user interface includes three menu styles.

- The icon menu style has large buttons that lead you into the different parts of the system and show you 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 the traditional character terminals.

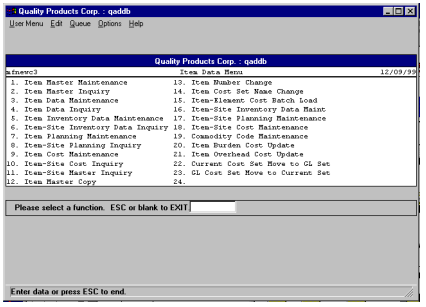
You can set up a user-specific choice of menu style in User Interface Profile.

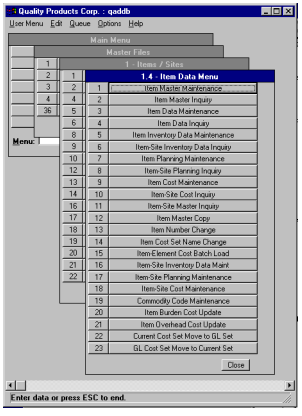
Fig. 11.8
Menu Style Options (GUI Interface)

Icon Menu



Character Menu





Tear-Off Menu

Assigning Security to OBCM Programs

Use Window Config and Security Maint (36.20.5) to set user access to the OBCM programs, folders, and fields. You can set program security to a level lower than the initial definition, change folder tab labels, and determine the sequence in which folders appear.

Window Config and Security Maint cannot increase a user's access level; it can only decrease access levels. Access, folder tab labels, and folder sequence are initially defined when the program is developed, either the original program developed by QAD or a program developed or modified by the customer. Programs are developed and modified using the Component Configurator tool set (36.20.13).

To determine the components and version information for a currently running OBCM application, choose About… from the Help menu and select the Revision button. The Program Version window displays.

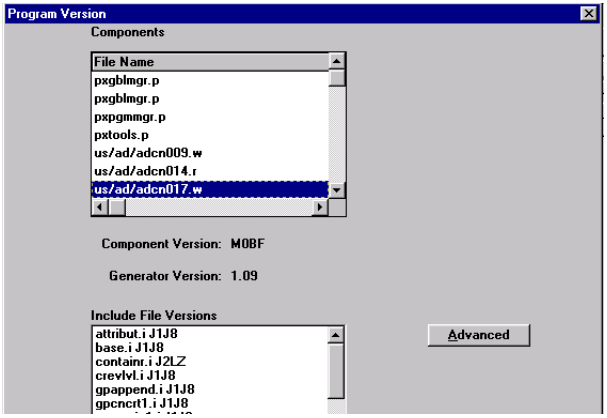


Fig. 11.9 Program Version Window

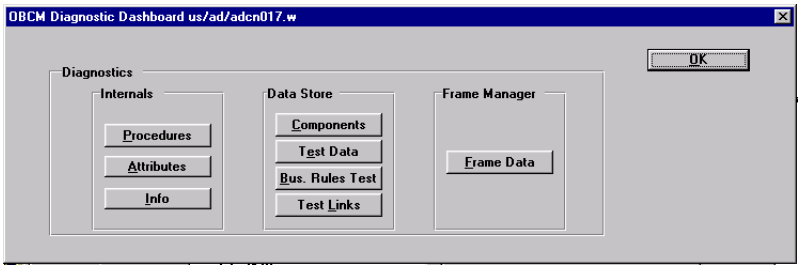
When you select a file in the Components list, its version information displays. A list of include file and versions also displays.

Use the Advanced button to view internal components of the current application and use the advanced diagnostics tool set. To enable this feature, you must edit your Windows-client initialization file and add the following line to the section titled [MFGPRO].

```
dashboard=yes
```

Do this prior to starting MFG/PRO eB. Once you have done this, click Advanced to display the OBCM Diagnostic Dashboard.

Fig. 11.10
OBCM Diagnostic Dashboard



Security Levels

OBCM programs include three levels of user access:

- Write access allows a user to update fields. This is the default level for most programs.
- Read-only access allows a user to see the program's fields but not update them.
- No access denies the user the ability to update or see the fields.

For read-only and no-access levels, security is downward effective. For example, if a program has read-only access, then all folders and fields within that program are also read only. Write-access security can be overridden at a lower level so that users have read-only or no access to fields within a write-access folder.

See "Using Menu Security" on page 40.

You set write or read-only access for programs with Window Config and Security Maint. To give a user a no-access security level to a program, use Menu Password Maintenance (36.3.1).

Folders and fields can have write, read-only, or no-access security.

Setting Window Read-Only or Write Access

1 In the Security folder, enter the container ID (the program name) for which you want to set access for the user you enter in Userid/Group. The user can be an individual or group. Leave the field blank to apply the security setting to all users.

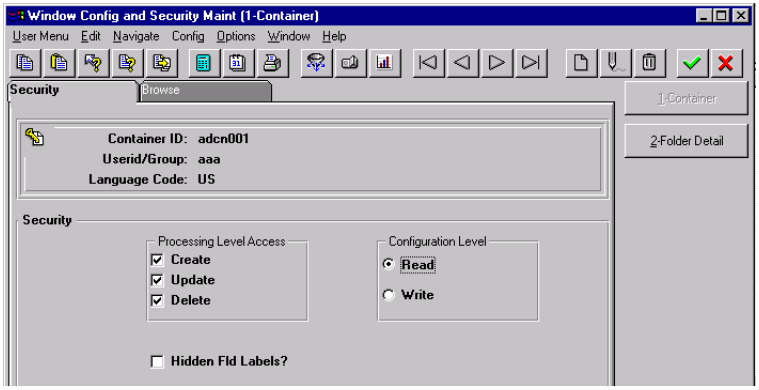


Fig. 11.11 Window Config and Security Maint (36.20.5), Security Folder

2 Enter the language code.

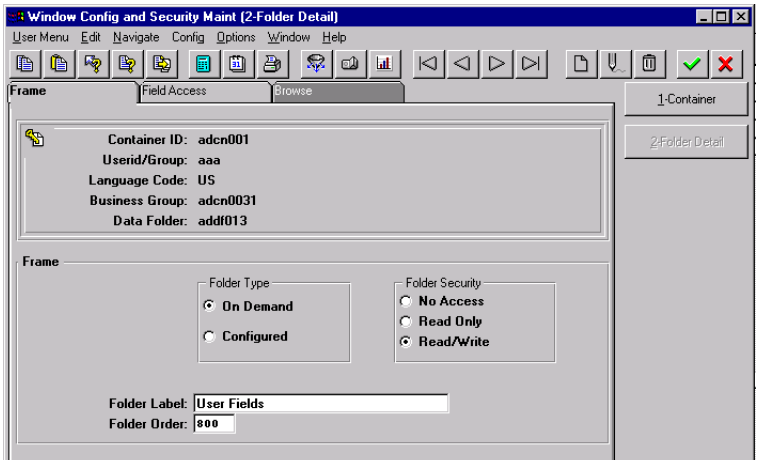
3 Select the processing-level access. Choose whether the user can update or delete records. If you do not select any access, the user has read-only access.

4 Select a Read or Write configuration level. Selecting Write lets a user configure an individual version of the program.

5 Check Hidden Fld Labels? to hide from users the fields they cannot access.

6 Select 2-Folder Detail to set security at the folder level.

Fig. 11.12
Window Config and Security Maint, Frame Folder



7 Enter the Business Group, which is a set of related folders within a program. Business groups control the initial definition of access levels. They are defined in the Groups application of Container Maintenance (36.20.13.13).

8 In Data Folder, enter the file name of the OBCM data folder whose access you want to restrict.

9 For Folder Type, select Configured for folders that you want to automatically display when users cycle through folder tabs. Select On Demand for folders that you want users to select for themselves by clicking on the tab or by selecting them from the Navigate menu. For example, in Figure 11.12, the Browse folder is an on-demand folder, which is indicated by its dark gray tab.

10 Select the level of folder security.

11 In Folder Label, enter or edit a folder label, which can be up to 24 characters. This appears on the folder's tab and the Navigate menu.

12 For Folder Order, enter an integer value greater than 10 for the order in which you want this folder to appear in relation to the other folders in the program. The value of 10 is assigned to the key folder, which must appear first.

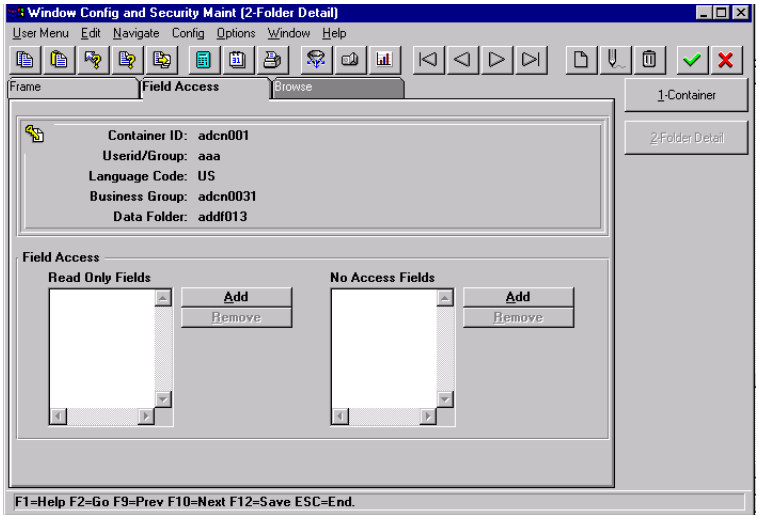


Fig. 11.13
Window Config and Security Maint, Field Access Folder

13 Select the Field Access folder of 2-Folder Detail to set security at the field level.

14 To enter all the field names you want to be read only, select the Add button to the right of the Read Only Fields list and enter the field names in the Add List Item dialog box, separating them with commas.

15 Choose Run to add them to the list.

16 To enter all the field names you want to be no-access, select the Add button to the right of the No Access Fields list and enter the field names in the Add List Item dialog box, separating them with commas.

17 Choose Run to add them to the list.

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 (in other words, the linking comparisons) between fields in different tables.

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 password 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 (36.20.13.12) to create or modify views.

You use some PROGRESS syntax in creating or modifying views. You must also understand MFG/PRO eB table and field relationships.

To create or modify a view:

1. Select the table or tables to include in the view.
2. Join the tables using PROGRESS logic.
3. Select fields from the tables.
4. Save the view.

Figure 11.14 and Figure 11.15 display graphic examples of how to create a view of selected fields from one table or two.

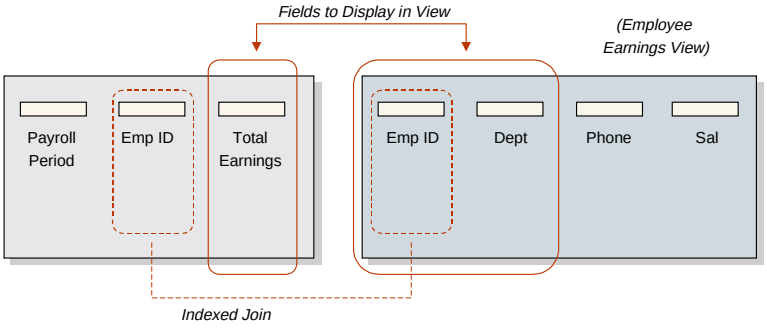


Fig. 11.14
Creating a View by Joining Two Tables

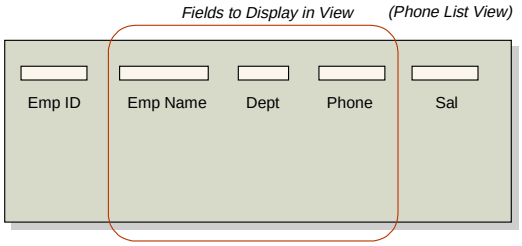


Fig. 11.15
Creating a View from One Table

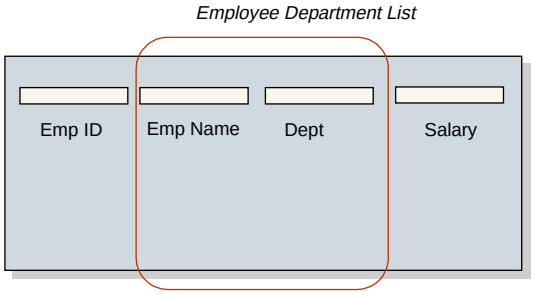
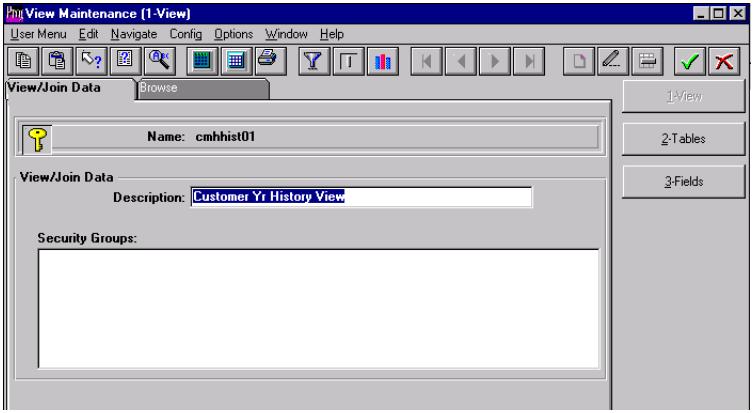


Fig. 11.16
View Maintenance (36.20.13.12), View/Join Data Folder



1 In the first window, 1-View, select or enter a view name in Name.

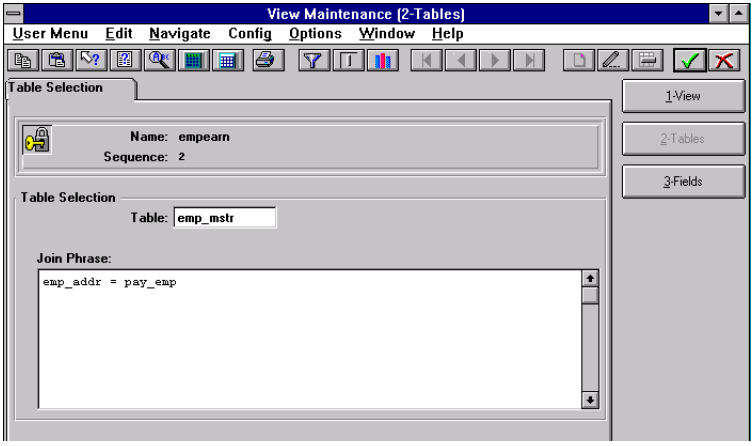
2 Press Go.

3 Enter a description of the view.

4 In Security Groups, enter a user ID to limit user access to the view (optional). You can enter multiple user IDs by separating them with commas.

5 Select the 2-Tables button.

Fig. 11.17
View Maintenance, Table Selection Folder



6 The number you enter in Sequence controls the order in which the table defined in Table is joined to the view.

7 Press Go.

8 Enter or edit the phrase to join the tables. Use proper PROGRESS syntax. Do not include a Where verb. Join phrases express the field relationships between tables (see Figure 11.14). For a faster display of fields, use indexed fields in the Join Phrase.

9 Select the 3-Fields button.

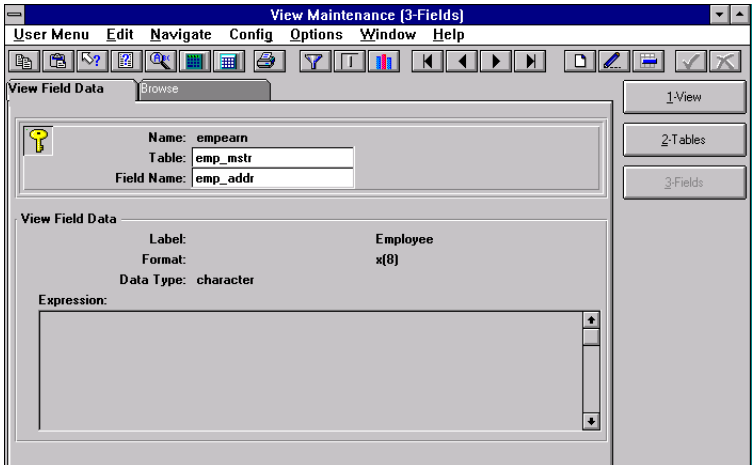


Fig. 11.18 View Maintenance, View Field Data Folder

10 In Field Name, enter a field from one of the tables in the view or enter a local variable. When entering a local variable, name it `local-varnn`, where `nn` is a number incremented by one from the last defined variable.

For example, you see from the look-up browse that the last local variable was `local-var05`; you name your local variable `local-var06`. Use local variables when you want to return a value resulting from an operation on two fields; for example, the quantity required minus the quantity open. Define the operation in Expression.

11 If you entered a local variable in Field Name, enter its Label, 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 eB session if you attempt to use the view.

13 Choose Run to save your changes.

Creating Browses

Use Browse Maintenance (36.20.13.10) 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 whose 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 letters from your name, then the extension `.p`.

For example, you create a power browse and name it `123456`; the system names the code `12br3456.p`. If you selected both power and look-up browses, the system generates two source-code files: `12br3456.p` and `12lu3456.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. Select the PROGRESS editor from the user menu to compile the browse.

Note You can access the PROGRESS editor from the user menu only if your propath is correctly set up to access source files.

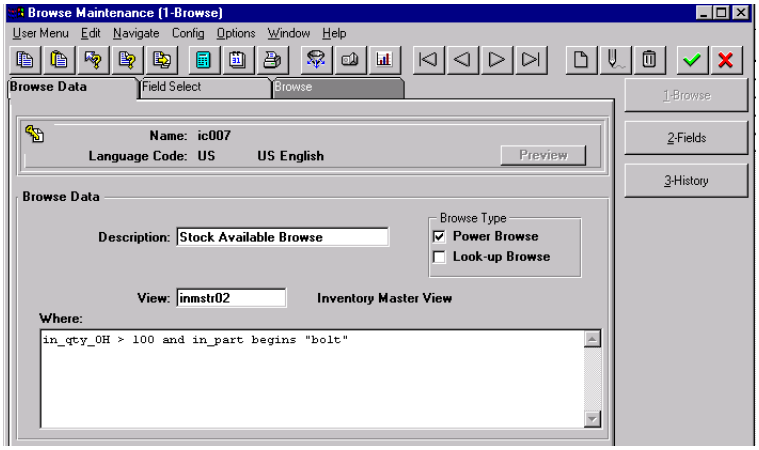


Fig. 11.19 Browse Maintenance (36.20.13.10), Browse Data Folder

To create or modify a browse:

1. In the fields of the Browse Data key folder, name the browse and select a Language Code. 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 eB module mnemonics or make up your own.

2. Press Go or select the edit icon to navigate to the data folder.
3. Enter a description. This description appears above the browse window, at the top of the on-demand report window, in the Drill Down description, and in the Menu Substitution label.
4. 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 (views where your user ID is in the password list). 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.

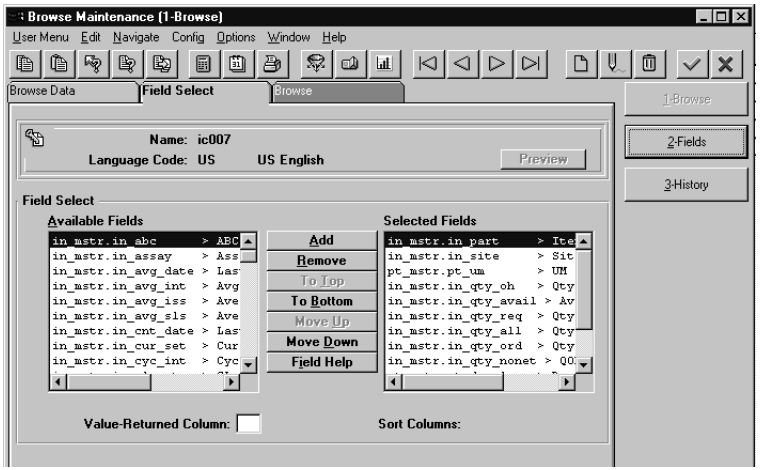
Note You must have already defined in View Maintenance the view name you enter in View, or you must enter a primary table name.

5. In the Where field, type the selection criteria (optional) to limit the browse's search to records that meet a certain condition. The criteria in Figure 11.19 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.

If you are using MFG/PRO eB in character mode, go to step 12.

6 In the Windows interface, select the Field Select folder.

Fig. 11.20
Browse Maintenance, Field Select Folder



7 Fields from the view or primary table, which you entered in the Browse Data folder, appear in Available Fields. Include up to 20 fields in your new browse by clicking on fields and choosing the Add button.

8 If you want to remove a field from the Selected Fields list, select it and choose the Remove button.

9 You can use the To Top, To Bottom, Move Up, and Move Down buttons to arrange the fields in the Selected Fields list or you can order the fields using the Value-Returned Column. Use the Value-Returned Column to enter the column number whose field value you want entered in the active field of the calling program. It must be a number from 1 through 7. The default is the first column of the browse.

When you have arranged the fields in the order you want, press Go.

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

10 In the Sort Columns field (in the Field Select folder), enter the columns that you want to have available to sort on. 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.

11 The look-up browse sorts the records on the first column you enter in the Sort Columns field. The remaining columns you enter are listed in the selection list above the browse. Select another column in the list and the browse re-sorts on that column. When it re-sorts, the browse redisplays beginning at the first record. The browse does not redisplay beginning at the record that was selected when the re-sort was initiated. By default, the browse sorts on the first field in the Selected Fields selection list.

Note Instead of the 2-Fields, Field Select folder, you must use the 2-Fields, Browse Field Data folder in character mode. The Browse Field Data folder is useful in the Windows interface to control the sequence of the tables and to alter the labels and format.

If you are using the Windows interface, go to step 13.

12 In character mode, select the 2-Fields button.

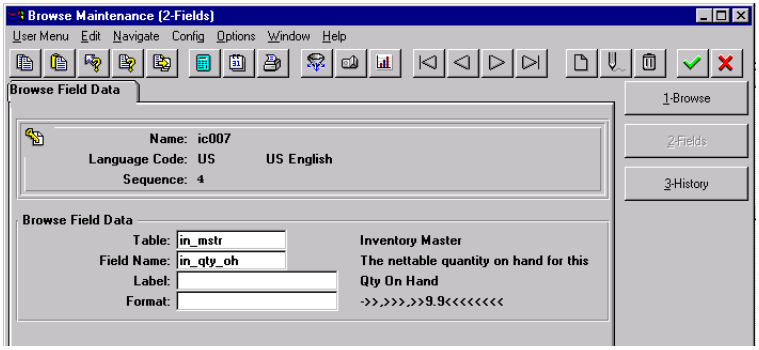


Fig. 11.21 Browse Maintenance (2-Fields)

13 Identify the table and field, type in the display format you want (character, integer, decimal, logical, or date), sequence of field, column in the display, and change the default field label and format.

14 Select the 1-Browse button.

15 Click on the Preview button to look at the new browse.

Important Previewing a browse can be a time-consuming process because the system generates and displays the browse in runtime.

16 If the fields, column labels, and data of the browse are acceptable, press Go. If the browse is not acceptable, return to the Field Select folder or 2-Fields application to rearrange the fields.

17 Choose Run to save.

Index

R

S

T

U

V

W