

MFG/PRO® eB

eB System Administration Training Guide

PROGRESS DATABASE
ON UNIX SERVER



70-2824A
MFG/PRO eB
April 2001

This document contains proprietary information that is protected by copyright. No part of this document may be photocopied, reproduced, or translated without the prior written consent of QAD Inc. The information contained in this document is subject to change without notice.

QAD Inc. provides this material as is and makes no warranty of any kind, expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. QAD Inc. shall not be liable for errors contained herein or for incidental or consequential damages (including lost profits) in connection with the furnishing, performance, or use of this material whether based on warranty, contract, or other legal theory.

Some states do not allow the exclusion of implied warranties or the limitation or exclusion of liability for incidental or consequential damages, so the above limitations and exclusion may not be applicable.

PROGRESS® is a registered trademark of Progress Software Corporation. Windows™ is a trademark of Microsoft Corporation.

MFG/PRO® is a registered trademark of QAD Inc. QAD, QAD eQ, and the QAD logo are trademarks of QAD Inc.

All other products and company names are used for identification purposes only, and may be trademarks of their respective owners.

©Copyright 2001 by QAD Inc. All Rights Reserved.

70-2824A

QAD Inc.
6450 Via Real
Carpinteria, California 93013
Phone (805) 684-6614
Fax (805) 684-1890

LearningServices@qad.com
<http://www.qad.com/services/learn/>

Contents

- PREFACE..... 7**
 - Using This Guide 8
 - Audience 8
 - Location of MFG/PRO Manager Functions Information 9
 - Organization 9
 - Document Conventions 10

- CHAPTER 1 COMPONENTS AND CONFIGURATION..... 11**
 - UNIX Basics 14
 - Configuration Basics 17
 - Client/Server Configuration 20
 - Batch Mode Client 22
 - PROGRESS Basics 23
 - PROGRESS Executables 27
 - PROGRESS Parameters 28
 - PROGRESS Database Architecture 29
 - MFG/PRO Basics 33
 - MFG/PRO Code 34
 - MFG/PRO Empty Databases 36
 - Default Databases 37
 - Main MFG/PRO Databases 38
 - Other MFG/PRO Databases 39
 - Custom Side Databases 40

CHAPTER 2	SYSTEM SIZING AND INSTALLATION	41
	Disk Space and Other Requirements	43
	Installation Overview	50
	PROGRESS Server Installation	51
	MFG/PRO Server Installation	54
	MFG/UTIL Basics	55
	Database Sets	57
	Server Media File Types	60
	Directory Structure on the Database Server	62
	Load or Update Online Help	63
	PROGRESS Client Installation	66
	MFG/PRO for Windows Installation	67
	MFG/PRO Icons	70
	Sample Parameter File	71
	Sample Hosts File	72
	Sample Services File	73
CHAPTER 3	DATABASE MANAGEMENT	75
	Data Movement	77
	Database Backups	79
	Delete and Archive	83
	Database Fragmentation	85
	Dump and Load	89
	Dump Procedure	91
	Load Procedure	93
	Bulk Load and Index Rebuild	95
	Special Dump/Load	98
CHAPTER 4	PERFORMANCE TUNING	101
	Tuning Techniques	103
	PROGRESS Parameters	104
	Configuring the BI File	107
	PROGRESS Performance Processes	108
	PROGRESS Monitor (Promon)	110

Using Promon to Set -B Parameter	111
CHAPTER 5 MULTI-VOLUME DATABASE	113
Multi-Volume Overview	115
Multi-Volume Components	116
Multi-Volume Creation Steps	117
Creating a Structure File	118
Sample Structure File	119
Determining Database and Extent Size	120
Calculating the Structure File Extent Size Number	121
Using Prostrct	122
Creating a Void Multi-Volume Database	123
Populating a Void Multi-Volume Database	124
Copying a Populated Database to a Void Multi-Volume Structure ..	125
CHAPTER 6 SYSTEM UTILITIES	127
Database Set Administration	129
Compiling Programs	130
PROGRESS Batch Job	133
Using UNIX cron	134
Establishing a Batch Mode Client Session	136
PROGRESS Batch Program	137
Batch Input File	138
APPENDIX A USER INTERFACE	141
User Interface Overview	142
Browsets	150
What Is A Browse?	150
Types Of Browsets	151
Look-Up Browsets	151
Power Browsets	152
Designing A Browse	154
Exercise 1	156
How To Create A Browse	157

VI EB SYSTEM ADMINISTRATION — TRAINING GUIDE

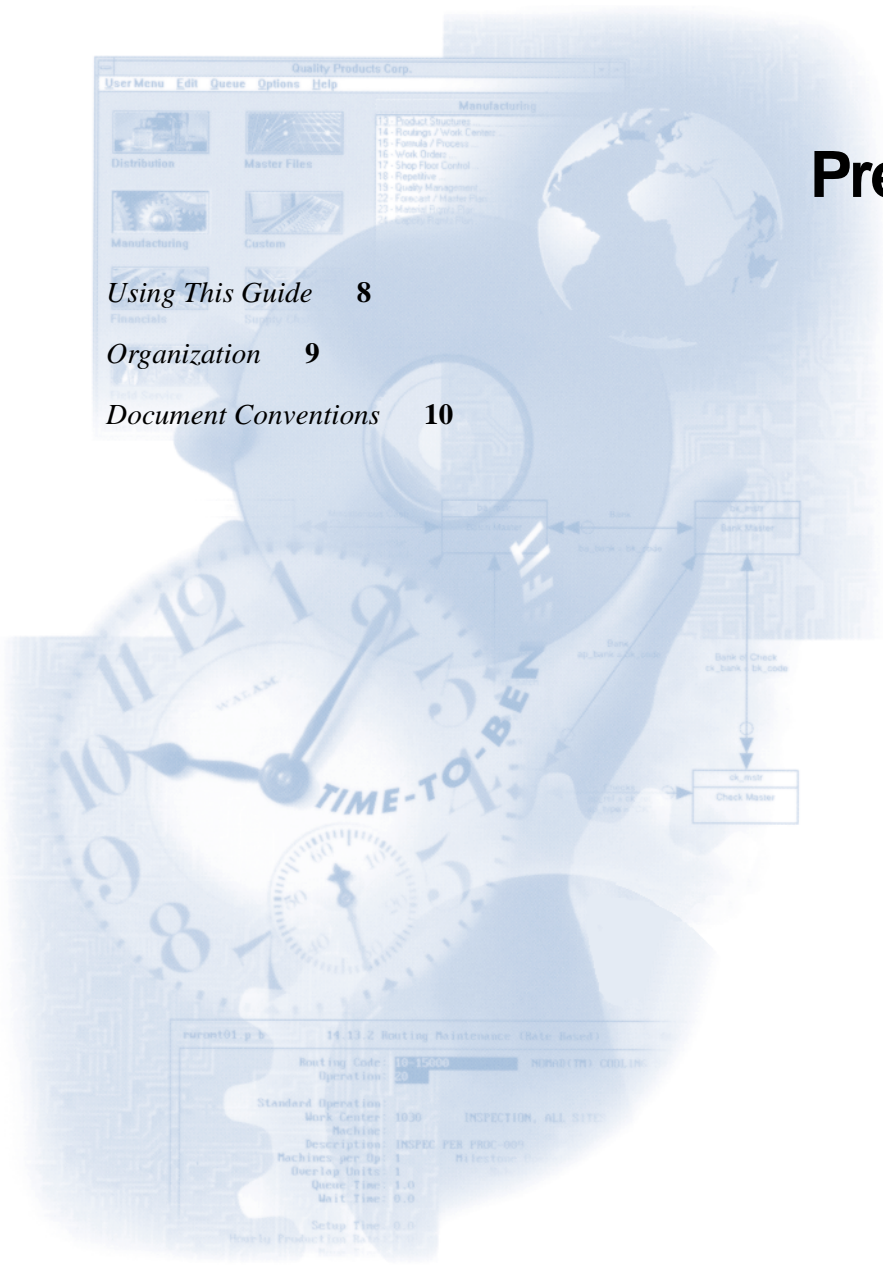
A Note about Browse Names	164
Exercise 2	165
Linking Your Browse	166
Linking a Look-Up Browse	168
Linking a Power Browse	169
Exercise 3	170
Views	171
What Is A View?	171
Designing A View	172
How To Create A View	173
Exercise 4	177
Character Environment	180
Exercise 5	185
Additional Information	186
Example Uses Of A Power Browse	186
Browse Graphing	187
Filtering and Sorting Browsers	191
Printing your Browse	196
Drilldown Maintenance	199
User Tool Maintenance	200
Window Config and Security Maintenance	201

Preface

Using This Guide **8**

Organization **9**

Document Conventions **10**



System Administration for MFG/PRO on UNIX

Includes the following:

- Components and Configuration
- System Sizing
- Installation and Configuration
- MFG/PRO Manager Functions
 - Control files, security, help, printers, batch, CIM
- System Management
 - Database management, performance and tuning, and administration utilities

Using This Guide

Use this guide for training information on MFG/PRO system administration, especially database management and related tasks.

Audience

These instructions are for the MFG/PRO system administrator who manages the MFG/PRO database and is familiar with UNIX and networking.

Location of MFG/PRO Manager Functions Information

Although it is involved with system administration, this guide does not include documentation for the MFG/PRO Manager Functions (menu 36). For that information, refer to the *Managers Functions Training Guide*

Organization

This guide is organized as follows.

1	Components and Configuration	Explains the basic concepts of PROGRESS and the MFG/PRO system.
2	System Sizing and Installation	Explains basic MFG/PRO hardware and software requirements and installation procedures.
3	Database Management	Explains the common database management tasks, along with some step-by-step instructions.
4	Performance Tuning	Explains options on how to boost PROGRESS and MFG/PRO system performance.
5	Multi-Volume Database	Explains how to set up a multi-volume database for optimal performance.
6	System Utilities	Covers various utilities that support the MFG/PRO application, particularly batch processing.
7	System Administration Labs	Contains labs that reinforce classroom and self-study activities.

Document Conventions

This guide covers both Version eB of MFG/PRO.

Command prompts use the conventions listed in the following table.

If you see:	It means:
monospaced text	A command or file name.
italicized monospaced text	Italicized monospaced text indicates a variable name for a value you enter as part of an operating system command. For example, <i>YourCDROMDir</i> .
indented command line	A long command that you enter as one line (although it appears in the text as two lines).

Components and Configuration

<i>UNIX Basics</i>	14
<i>Configuration Basics</i>	17
<i>Client/Server Configuration</i>	20
<i>Batch Mode Client</i>	22
<i>PROGRESS Basics</i>	23
<i>PROGRESS Executables</i>	27
<i>PROGRESS Parameters</i>	28
<i>PROGRESS Database Architecture</i>	29
<i>MFG/PRO Basics</i>	31
<i>MFG/PRO Code</i>	32
<i>MFG/PRO Empty Databases</i>	34
<i>Default Databases</i>	35
<i>Main MFG/PRO Databases</i>	36
<i>Other MFG/PRO Databases</i>	37
<i>Custom Side Databases</i>	38

System Administration for MFG/PRO on UNIX

Includes the following:

- Components and Configuration
- System Sizing
- Installation and Configuration
- MFG/PRO Manager Functions
 - Control files, security, help, printers, batch, CIM
- System Management
 - Database management, performance and tuning, and administration utilities

UNIX Basics

- Command syntax contains option flags
 - Example: `ls -l` (list files in long format)
- UNIX files have an owner and group assignment
 - Change file ownership using the `chown` command
 - Example: `chown sra start.Demo`
 - Files have privileged groups, changed by `chgrp`
 - Example: `chgrp mfg start.Demo mfg`

UNIX Basics

- UNIX files are permission protected
 - Set for owner, group, and others using `chmod`
 - Example: `chmod 777 start.Demo`
- VI is the default editor
 - Example: `vi start.Demo`
- View processes in memory with `ps`
 - Example: `ps -ef`
- Commands can be passed to filters
 - Example: `ps -ef | grep _mprosrv`
- Command output can be redirected
 - Example: `ls > file.out`

UNIX Basics

To manage MFG/PRO, you must know how to use some basic UNIX commands. Refer to the following tables as needed during this training course. Refer to your operating system manuals for more information.

Table 1.1
Basic vi Editor Commands

Action	Command
Left	h
Down	j
Up	k
Right	l
Enter command mode	ESC
Insert text	i
Delete one character	x
Write (save) file	:w
Write and quit	:wq
Quit without write	:q!

Table 1.2
Basic UNIX Commands

Action	Command
Directories	
Display directory path	pwd
Change directory	cd
Change back one directory	cd ..
List files in directory	ls
List files (long format)	ls -l
List files (all)	ls -a
File Permissions	
Change permission	chmod <i>Permission Value</i>
Change group	chgrp <i>GroupName FileName</i>
Change owner	chown <i>UserID FileName</i>
Help	
Command specific help	man <i>Command</i>

Table 1.3
Other Commands

Action	Command
Copy file	<code>cp OldName NewName</code>
Open vi editor	<code>vi FileName</code>
Search for lines in a file containing a given text string	<code>grep TextString FileName</code>
Search for files whose names contain a given text string	<code>ls grep 'TextString'</code>
Display processes	<code>ps</code>
Display all processes (long format)	<code>ps -ef</code>

Configuration Basics

- MFG/PRO can run in the following configurations
 - Host-based mode
 - Client/Server mode
- On a UNIX database server, host-based mode is the default
 - Note: A batch client is also available, which runs in a non-interactive background mode
- Client/Server mode is used to run MFG/PRO in a Windows GUI interface

Configuration Basics

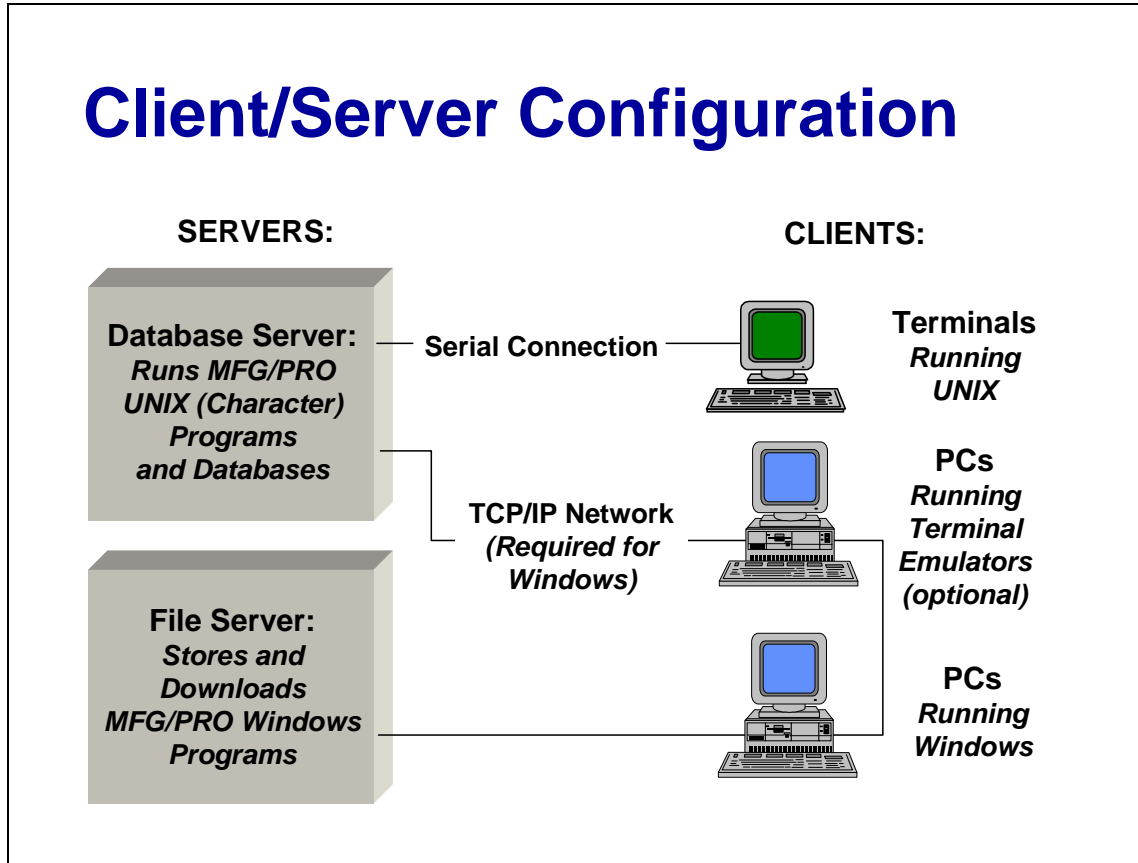
The next few slides outline configuration concepts both generally and specifically to MFG/PRO.

Host-Based Configuration


- Host mode means the client process runs on the same physical processor(s) as the database server process
- MFG/PRO sessions run on dumb terminals or terminal emulators
- MFG/PRO connects directly to the database via a local directory path

Client/Server Configuration

- Client/server means the client process runs on a separate processor than the database server process
- MFG/PRO for Windows (GUI) sessions run on PCs
 - Programs execute on the PCs but are typically stored on a network file server
- MFG/PRO connects to the database through a TCP/IP LAN network
- Clients can use either the graphical user interface (GUI) or character-based interface



Client/Server Configuration

 See *MFG/PRO Gui Client Configuration* in the *MFG/PRO eB Installation Guide, Progress on Unix*

Host-Based and Client/Server Comparison

	Host Client:	Windows Client/Server:
CPU:	Database Server	PC
Connection Types:	Single user or multiuser	Multiuser only
Connection Parameters:	Requires DB directory path	Requires network settings in the Hosts and Services files
Session Startup:	Uses O/S script	Uses icons, initialization file (.ini), and parameter file (.pf)
Session Interface:	Character only	Windows GUI or Windows Character

Batch Mode Client

- MFG/PRO supports batch mode clients
- Batch mode clients access the database server through a direct, shared-memory connection
- Batch mode clients invoke MFG/PRO functionality without an interactive client session
- Any needed input/output is read from, or sent to, respective files
- Batch clients access the character-style programs on the database server

Batch Mode Client

For information on scheduling MFG/PRO processes through the batch mode client, refer to Chapter 6, “System Utilities” in this training guide.

PROGRESS Basics

- PROGRESS is both a RDBMS and 4GL programming language
 - Standard SQL commands can also be used
- The PROGRESS installation directory is referred to as *ProgressDir*
 - To identify version: `cat ProgressDir/version`
 - PROGRESS version 9.1B as of Mon Jan 8, 2001
 - To list installed products:
`cat ProgressDir/progress.cfg`
- PROGRESS offers various products/packages:
 - Database server products/packages
 - Client and network connection products
 - Development and administration products/packages

PROGRESS Basics



See *Progress Basics* in the *Progress System Administration Reference, Progress Version 9*

Database Server Products/Packages

- Enterprise Database Server
 - Contains RDBMS, server networking, and minimal (Runtime) client product
- Workgroup Database Server
 - Package is similar to Enterprise Database Server, but without key performance aids such as Before-Image Writers
 - Not recommended for production systems

Client Products

- Client Networking
 - Enables clients to access remote databases running with the proper network parameters
 - Runtime client, which executes compile 4GL programs
- Query
 - Tool for writing inquiries and reports (only compiles programs that read, not write)
 - Required to run MFG/PRO browses

Development and Administration Products/Packages

- 4GL
 - Available on Windows and UNIX
 - Compiles any program, including those that write to the database
 - Provides database administration utilities
- ProVISION package
 - Available on Windows only
 - Includes 4GL and other development tools
- At least one license of 4GL or ProVISION required for MFG/PRO installation and administration

PROGRESS Executables

- PROGRESS has pre-generated executables to perform tasks such as starting servers and client sessions
- Executable files typically reside in the *ProgressDir/bin* sub-directory
- The executable files typically begin with an underscore
 - Example: `_mprosrv` (starts a multiuser server)

PROGRESS Executables



See *Progress Executables* in the *Progress System Administration Reference, Progress Version 9*

PROGRESS Parameters

- Parameters instruct the executables how to behave, such as how to handle hardware
 - Example: `_mprosrv DbName -B 2000`
(starts a server with specified buffer memory)
- Parameters are specific to the PROGRESS executable(s)
 - Example: -B only applies to `_mprosrv`
- Each parameter takes affect at a specific point, such as program startup

PROGRESS Parameters



See *Progress Parameters* in the *Progress System Administration Reference, Progress Version 9*

PROGRESS Database Architecture

- As of Progress Version 9 all databases are multi-volume in structure
 - Note: For production, a multi-volume database should be constructed to distribute I/O across multiple channels (covered later in this training)
- Databases have a structure of logical files called a *schema*
 - A schema defines data storage and consists of:
 - Tables (also called files): logical groupings of data, such as customer information
 - Fields (called columns in other systems): the data elements within tables, such as address
 - Indexes: maps the location of data to tables and fields

PROGRESS Database Architecture



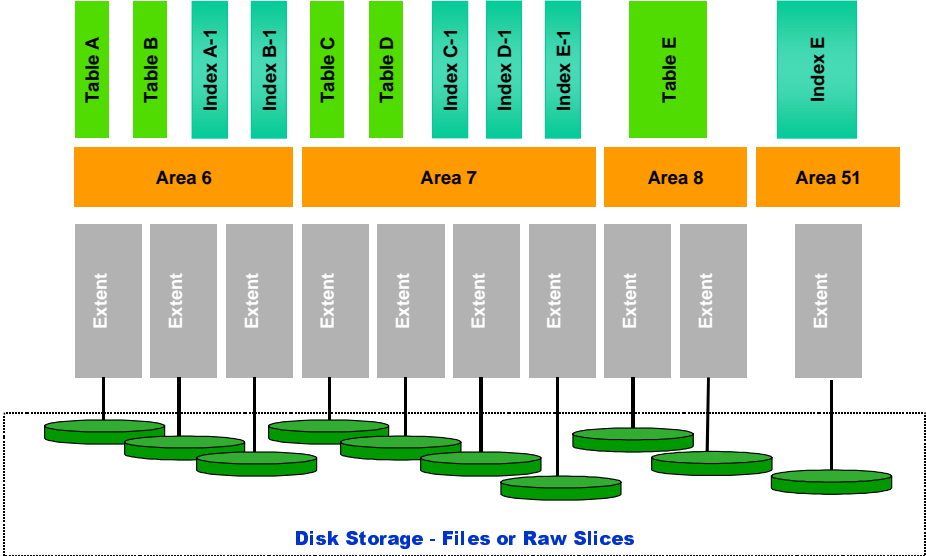
See *Progress Database Architecture* in the *Progress System Administration Reference, Progress Version 9*

PROGRESS Database Architecture

- Database logical files are created within specific Data Storage Areas during the database build
- Data Storage Areas are comprised of one or more physical files known as database extents
- Database extents are created in specified locations, and therefore can be directed to specific I/O channels.

PROGRESS Database Architecture

Progress V9 Database Layout - Area Architecture



PROGRESS Database Files

- All databases use roll-backward recovery
 - This Data Storage Area is the Primary Recovery Area
 - The extents within this area contain snapshots of data prior to each alteration
 - If a transaction is aborted, the data from the .bx extent is used to reset the database
- Other related files include:
 - Empty Database (empty.db)
 - Database Extent File (.d1, .d2, .dx, .a1, .b1)
 - Database Definitions (.df)
 - Data (.d)
 - Log file (.lg) and lock file (.lk)



See *Primary Recovery Area* in the *Progress System Administration Reference, Progress Version 9*

MFG/PRO Basics

- MFG/PRO is a Global Supply Chain Management software package from QAD Inc.
- MFG/PRO is written in PROGRESS 4GL
- Each MFG/PRO release requires a specific version of PROGRESS
- MFG/PRO programs come in modules, so you can purchase only the modules you need
- For version: `cat EBInstallDir/version.mfg`
 - Example: Release Eb01 as of Dec 3, 2000

MFG/PRO Basics



See *MFG/PRO Basics* in the *MFG/PRO eB User Guide Volume 1, Introduction*

MFG/PRO Code

- *Nonencrypted source code is:*
 - *Readable and modifiable*
 - *Stored as .p file in eBDir/ClientDir/LanguageDir/src*
- *Encrypted source code is:*
 - *Encoded and non-modifiable*
 - *A .p file in eBDir/ClientDir/LanguageDir/xrc*
 - *Maintenance programs are xrc, unless premium is paid*
- *Source code programs are compiled to produce executable code*
 - *Executable (object) programs are stored as a .r file*

MFG/PRO Code



See *Database Server Software* and *Gui Client Software* in the *MFG/PRO eB Installation Guide, Progress on Unix*

Void vs Empty Database

- Before we introduce the MFG/PRO databases, keep in mind the distinction between void and empty databases
- Void Database
 - Contains only a header block
 - Contains no schema
 - The PROGRESS empty.db database is void
- Empty Database
 - Copy of void database with schema file (.df) loaded
 - Empty databases are empty of data

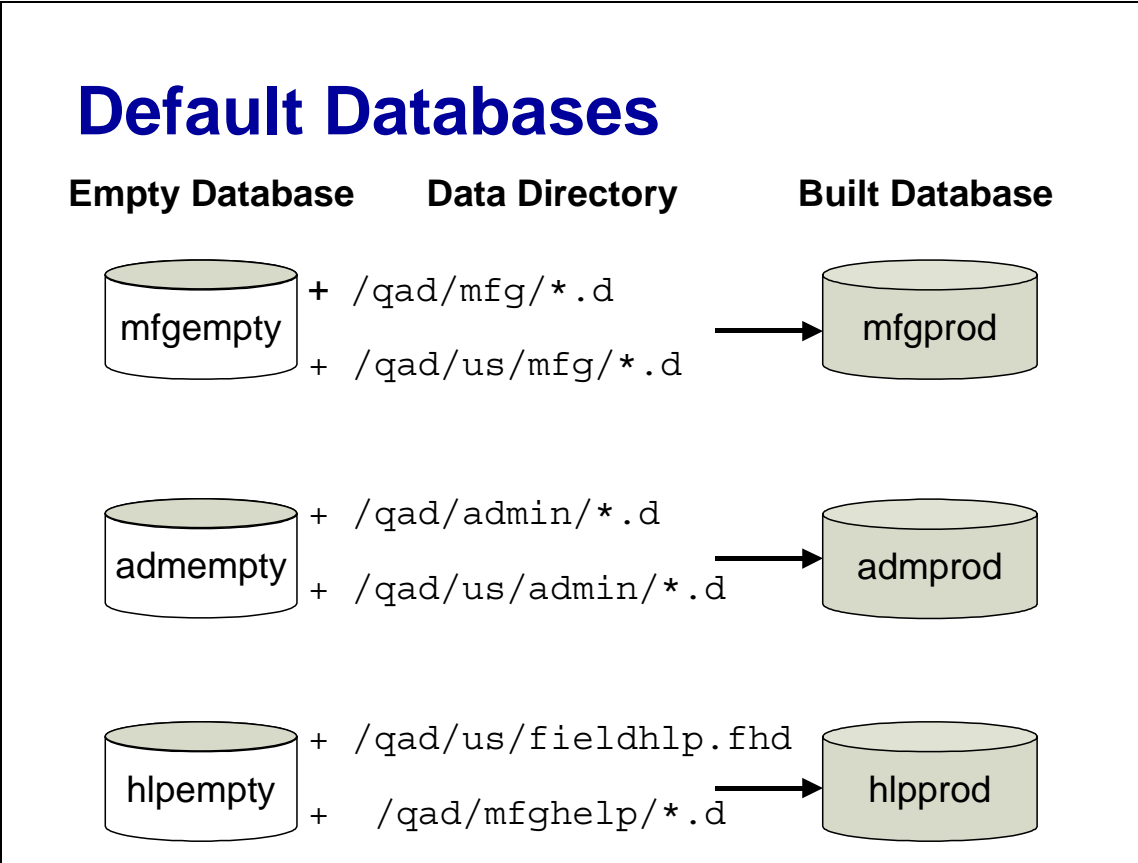
MFG/PRO Empty Databases

- MFGEMPTY is the empty, main MFG/PRO database.
 - Contains schema for most business data, such as customers, vendors, items, etc.
- ADMEMPTY
 - Contains schema for interface-related data (browses, drill-downs, views, etc.)
 - Schema for all the OBCM component data
- HLPEMPTY
 - Schema for field, procedure, and cross-reference help data

MFG/PRO Empty Databases



See *MFG/PRO Databases* in the *MFG/PRO eB Installation Guide, Progress on Unix*



Default Databases

- Copies of the empty databases are made and specific accumulations of predetermined data are loaded into these copies to populate them.
- This predetermined data is stored as series of data files (.d files), located predominately in sub-directories under the eB installation directory, with some language specific files elsewhere.
- There is one .d data file for each of the logical files created within the database when the schema (structure) file (.df) is loaded.
- All .d files are loaded into a database to produce the complete database contents.

Main MFG/PRO Databases

- Mfgprod
 - Empty except for language specific menus and messages
 - Used to make an empty production database
- *YourProductionDb*
 - Copy of mfgprod to be used as your live database
- Mfgtrain (Built separately in specific language)
 - Copy of mfgempty (or unused mfgprod) with training data loaded from the mfgtrain subdirectory
- Mfgdemo (Built separate in specific language)
 - Copy of mfgempty (or unused mfgprod) with demonstration data loaded from the mfgdemo subdirectory

Main MFG/PRO Databases

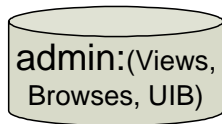


See *MFG/PRO Databases* in the *MFG/PRO eB Installation Guide, Progress on Unix*

Other MFG/PRO Databases

Database:

Related Application Item:



Employee Maintenance (1-Employee)

User Menu Edit Navigate Config Options Window Help

Personal Address Employment User Fields

Browse

Employee: BJV

Employee	Last Name
BJW	Whitehead
GLB	Bellicose
JRS	Smyth
JCW	Wilson
KWV	Underhill
...	...

Filter = Off

Field Help - Employee

... a specific employee—pointing to their name and address, employee data, and applicable employment information.

EFFECT: Whenever you add a new employee, you can enter the employee address code manually or let the system assign one. (If you leave employee code blank and press Enter, the next number will be assigned.) The system maintains this next number in the Employee Control File and increments it automatically.

NOTES:

IMPLEMENTATION-ISSUES: Employee names and addresses, unlike others (customers, suppliers, salespersons), are stored in a separate database file. Although employee numbers must be unique among themselves, it's okay if they overlap other address codes.

Some reports can be selected and sorted by address code, including some shop floor control, service/support, and payroll reports. This may influence your assigning address

Close

Other MFG/PRO Databases



See *Support Databases* in the *MFG/PRO eB Installation Guide, Progress on Unix*

Custom Side Databases

- If you customize MFG/PRO, you may need additional logical files or fields to store data manipulated by the custom program
- ***Never alter the standard MFG/PRO databases***
- Instead, copy and modify the PROGRESS empty.db to include your custom schema modifications
 - When a client runs the custom code, it must connect to the custom side database

Custom Side Databases



See *Support Databases* in the *MFG/PRO eB Installation Guide, Progress on Unix*

System Sizing and Installation

<i>Disk Space and Other Requirements</i>	43
<i>Installation Overview</i>	50
<i>PROGRESS Server Installation</i>	51
<i>MFG/PRO Server Installation</i>	54
<i>MFG/UTIL Basics</i>	55
<i>Database Sets</i>	57
<i>Server Media File Types</i>	60
<i>Directory Structure on the Database Server</i>	62
<i>Load or Update Online Help</i>	63
<i>PROGRESS Client Installation</i>	66
<i>MFG/PRO for Windows Installation</i>	67
<i>MFG/PRO Icons</i>	70
<i>Sample Parameter File</i>	71
<i>Sample Hosts File</i>	72
<i>Sample Services File</i>	73

System Sizing

- Disk
- Memory
- Database
- Network

Disk Space Requirements

- MFG/PRO Database Server Software
 - Approximately 1.25Gb (Without Character Code)
 - Approximately 800Mb (Install with Server Software)
 - Additional 800 Mb for each additional language
 - Additional 300 Mb if you load source code cross-reference help
- MFG/PRO Windows Client Software
 - 950Mb per language (including source code)
- PROGRESS Software (Per Installation)
 - 225 - 350Mb (depending upon installed products)

Disk Space and Other Requirements

The installation guide for your version of MFG/PRO details the minimum and recommended hardware and software requirements.



See *Disk Space Requirements* in the *MFG/PRO eB Installation Guide, Progress on Unix*

Memory Size Formula

- The memory size formula is:

$$[(6 \times \text{Number_of_Concurrent_Users}) + 32 + (\text{Number_of_DBs} \times 48)] \times 1.2 = \text{Mb}$$

- Example with 50 users and 2 production databases:

$$[(6 \times 50) + 32 + (2 \times 48)] \times 1.2 = 519.4\text{Mb}$$



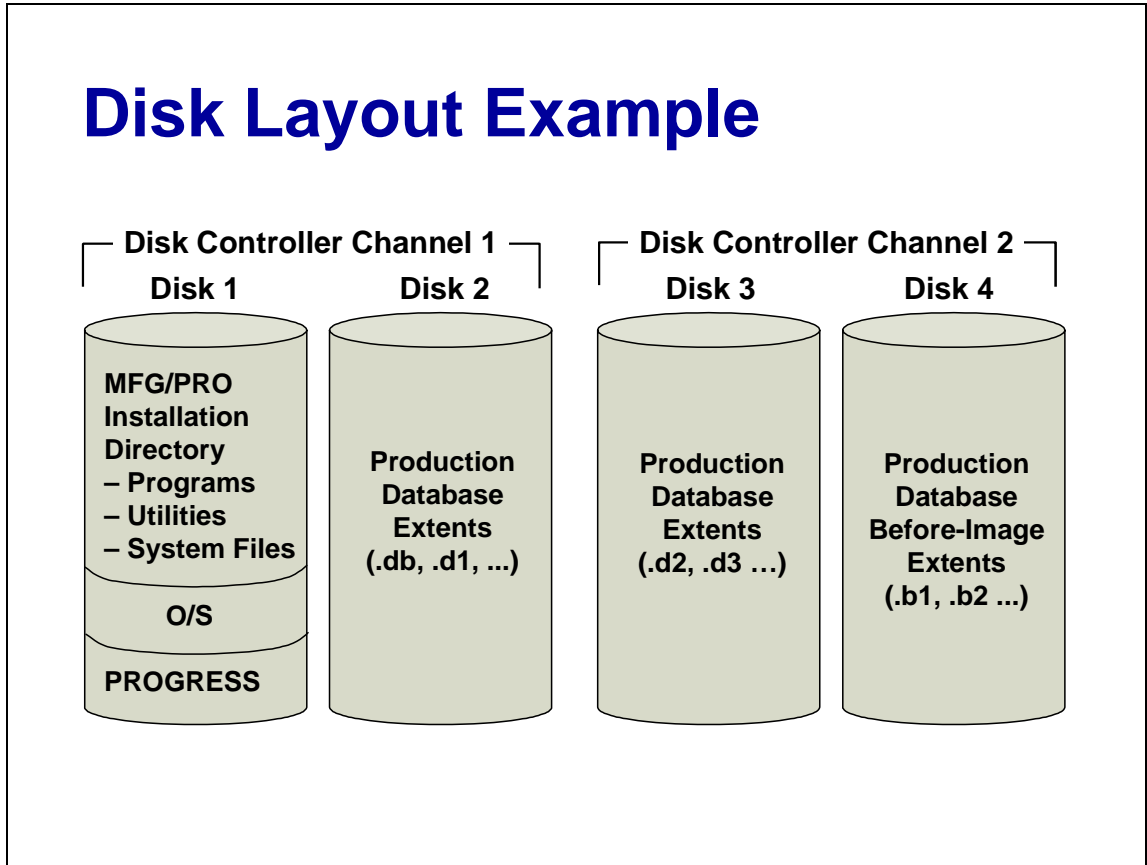
See *Memory Size Formula* in the *MFG/PRO eB Installation Guide, Progress on Unix*

Database Size

- Production database size varies greatly depending on business factors
 - 2 to 4Gb is a typical production database size
- For a new MFG/PRO installation, size the extents within each Data Storage Area to house the current/expected data assigned to each Area, while still considering overall database size.
- If converting from a legacy system, current database size must be considered



See *Database Size* in the *MFG/PRO eB Installation Guide, Progress on Unix*



- MFG/PRO installations with minimal users need a **minimum** of four separate disk drives for acceptable system performance.
- A **minimum** of two separate controller channels.
- More disk drives and controller channels will be required for medium to large systems.
- 2Gb drives are sufficient.



See *Disk Layout* in the *MFG/PRO eB Installation Guide, Progress on Unix*

Client PC Specs

- P233 or greater
- 64Mb RAM or greater
- 32-bit Network Interface Card
- 10–25Mb Local Disk Storage
- Super VGA Video

- Client must be able to “ping” the Database Server machine



See *Gui Client Hardware* in the *MFG/PRO eB Installation Guide, Progress on Unix*

Local Area Network Specs

- 100Mbps Backbone
- 10Mbps Shared-Media Segments
- Maximum 10 Clients per Shared LAN Segment (Depending on client activity)

- WAN use will require a supporting technology, and should be avoided



See *Network Design and Optimization* in the *MFG/PRO eB Installation Guide, Progress on Unix*

Installation and Configuration

Installation Overview

- Install server PROGRESS and server MFG/PRO on the database server machine and generate the database sets and system icons
- Install client PROGRESS on a file server or on each client machine (225-350MB/client)
- Install client MFG/PRO on a file server
- Establish the clients' network drives and set up each client
- Tailor Services & Hosts files on the server and client
- Set up database command scripts in ProControl
- **BACK UP!!!**

Installation Overview

The installation milestones shown above are explained in the following slides.



See *Installation Summary* in the *MFG/PRO eB Installation Guide, Progress on Unix*

PROGRESS Server Installation

- PROGRESS media contains all PROGRESS products
 - During installation, you type codes from the license addendum to specify which products to extract
- Server requires:
 - Enterprise database
 - At least one 4GL license

PROGRESS Server Installation

Installing PROGRESS on the server is a prerequisite to the MFG/PRO installation.

PROGRESS Installation Window

Product Configuration Data	
Company Name: _____	[Enter=Additional] [Ctrl-E=Done]
Serial Number: _____	[CTRL-T=Quit]
Release Number: 9.1B _____	CTRL-N=Release Notes]
Control Number: _____	[CTRL-V=View] [TAB=Next Field]

MFG/PRO for Unix Server Installation

- The Database Server installation is accomplished through a series of separate installation processes
- The entire compliment of database build files and executable code is spanned across several individual component CDs
 - Most Database Server files are on the Database Server CD
 - Language specific files are on the Language CD
 - Character executable code is on the Character Client CD
- Some CDs contain both Intel and Unix based install routines, and some are used in multiple steps.

MFG/PRO for Unix Server Installation

- Media is extracted, including database build files, language specific files, and object/source code
 - Code is character-based for use by non-interactive batch clients
 - Character style, telnet clients also access this code base
- An Wizard guides you through unloading and then calls the MFG/UTIL installation utility
 - MFG/UTIL is started only after the final CD is extracted
 - MFG/UTIL guides you through database creation and database set maintenance

MFG/PRO Server Installation

The server installation sets up the databases used for MFG/PRO sessions.



See *Database Server Setup* in the *MFG/PRO eB Installation Guide, Progress on Unix*

MFG/UTIL Basics

- MFG/UTIL is the installation and configuration utility
- MFG/UTIL is invoked by the install routine to:
 - Build your production databases via the “MFG/PRO Guided Setup” function
 - Configure database set specifics, including network parameters, to create start-up and shutdown icons
- After installation, use MFG/UTIL to administer database builds, database sets, scripts, and compilation

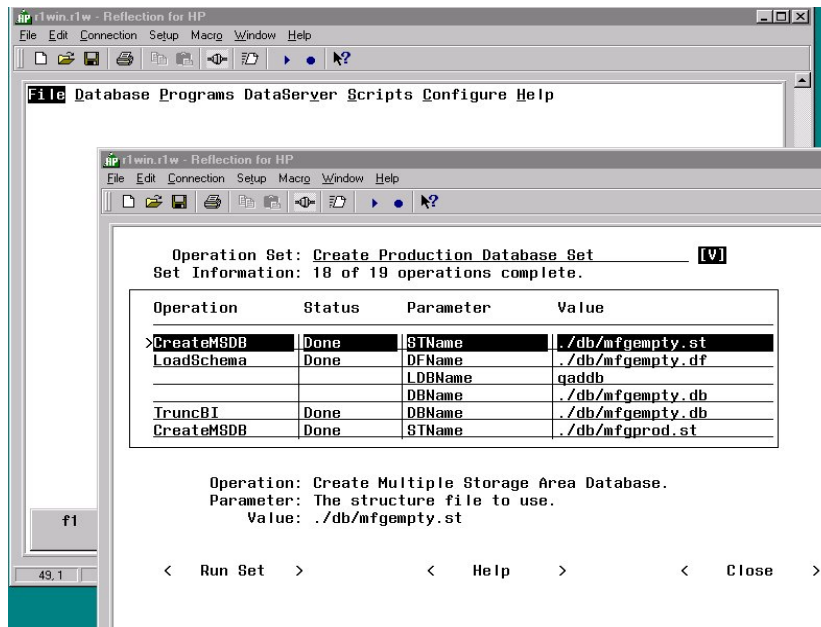
MFG/UTIL Basics

Because it is used throughout installation, you should become familiar with MFG/UTIL.



See *MFG/UTIL* in the *MFG/PRO eB Installation Guide, Progress on Unix*

MFG/UTIL Guided Setup



Database Sets

- MFG/PRO runs against a *set of databases*
 - Includes the main database, the administration database (user interface and OBCM data), the help database (on-line help), and and additional/custom databases required
 - You may have database sets for different purposes, such as production or training
- MFG/UTIL defines your database sets
 - Example: Training set may consist of the databases mfgtrain, hlpprod, and admprod
- MFG/UTIL generates system icons and .bat files from the set information
 - Used for database startup, shutdown, and client sessions
 - The resulting .bat filenames consist of set name and extension
 - Includes client/server parameters (-H, -S, -N)

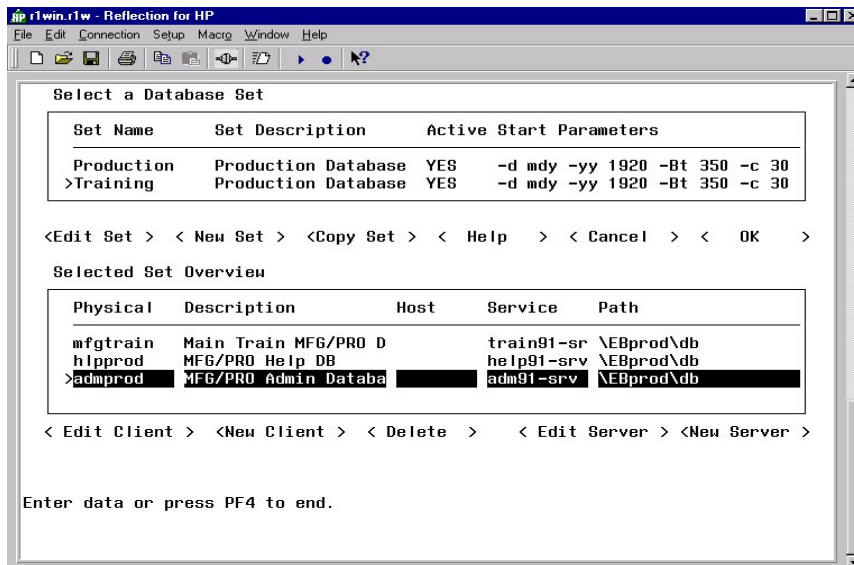
Database Sets

NOTE: You will edit a production set on the Host first which is saved back into MFG/UTIL. You then edit this set for the server databases if necessary, and then edit the client sets.



See *DataBase Sets* in the *MFG/PRO eB Installation Guide, Progress on Unix*

MFG/UTIL Database Set Window



Server Scripts

- When MFGUTIL completes, scripts are created (where xxxxxxxx = the set name)
 - Start .xxxxxxx
 - Starts a server process for each database in the set
 - Stop .xxxxxxx
 - Stops server processes for each database in the set
 - client .xxxxxxx
 - Starts a multiuser client MFGPRO session, connecting to all the databases in the set
 - single .xxxxxxx
 - Starts a single-user client MFGPRO session, connecting to all the databases in the set
 - qma .xxxxxxx
 - Required if Netui clients are used for maintenance functions
 - Spawns a hidden telnet session, used by remote clients, to invoke a character session

Server Media File Types

- Delivered files:
 - .p (source code)
 - .i (include)
 - .v (validation)
 - .o (object-oriented source code)
 - .t (trigger source code)
 - .w (widget source code)
 - .r (object code)
 - .df (database definition)
 - .d (data)
 - .qc (PROGRESS results join table)
 - .pl (procedure library)
 - .tpl (template)
- Generated files:
 - .db files/extents
 - scripts

Server Media File Types

Table 2.1
Files Delivered with Media

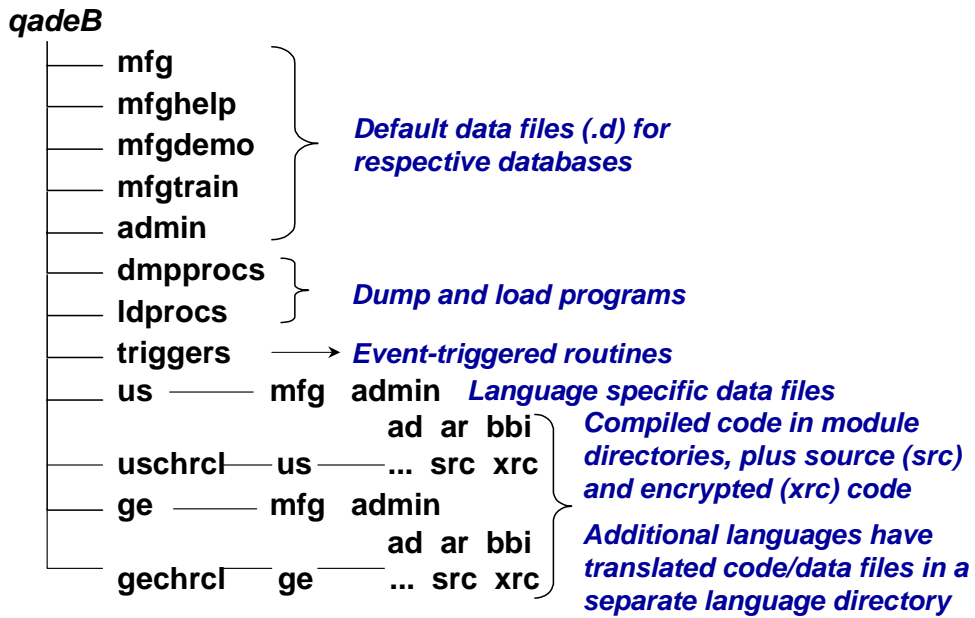
Extension	File Type	Description
.p	Source Code	Uncompiled MFG/PRO program code. The xrc sub-directory includes encrypted, uneditable source code. The src sub-directory includes any nonencrypted, editable source code you have purchased.
.i	Include	Subroutines referenced by the .p source code.
.v	Validation	Programs associated with logical fields for system security and validation logic.
.o	Object-Oriented Source Code	MFG/PRO source code created through the Object-Based Component Module (OBCM).

Extension	File Type	Description
.t	Trigger Source Code	Uncompiled database trigger code. Triggers are programs that launch when a certain database event occurs.
.w	Widget Source Code	Uncompiled programs relating to the Windows interface display.
.r	Compiled (Object) Code	Machine-executable MFG/PRO programs created from compiling the source code.
.df	Database Definition	ASCII text files that contain PROGRESS syntax for creating a database schema.
.d	Database Data	ASCII text files that contain field data, which you can load into a database with the appropriate schema.
.qc	PROGRESS Results Join Table	Table of relationships between database logical files for use in creating custom reports in the PROGRESS Results tool.
.pl	Procedure Library	Contains logic for several individual, interrelated program routines.
.tpl	Template	Template used to create the system .bat files that start and stop your databases.

Table 2.2
Other Database Files

Extension	File Type	Description
.db	Database	Contains the map to all the database extents, including the before-image and after image extents. A predominately empty database is built during installation. Copies of this database can be made to use as production, or to be loaded with training or demonstration data.
.dx	Database Extent	A physical file containing database data. A database may have several extents, physically located on a specific I/O channels. Each extent is part of a Data Storage Area.
.bx	Database Before-Image Extents	Contains data not yet committed to the database file. It is time-stamped and used in the event of a database recovery.
.ax	Database After-Image Extent	Contains data committed to the database since the last database backup. These files should reside on a separate disk from the database. You could use these files, in conjunction with the last backup, to restore the database to the point of the failure.
.lg	Database Log	Text file containing milestones of database activity, such as creation date, logins, and errors.
.lk	Database Lock	PROGRESS file created when the database server is running. Prevents database corruption.
.lic	PROGRESS License	PROGRESS file that controls the total number of users that can log into the PROGRESS database.

Directory Structure on the Database Server



Directory Structure on the Database Server

The above illustration shows the major MFG/PRO sub-directories but excludes miscellaneous directories.

Load or Update Online Help

- Start MFG/PRO session in single user mode
- Perform the *Field Help Load* (36.4.19)
- Enter the *language* (us) and *loadfile* (fieldhlp.fhd)
- Complete load

Load or Update Online Help



See *Loading the Field Help Data* in the *MFG/PRO eB Installation Guide, Progress on Unix*

Source Code Cross-Reference

- Optional load
- Requires additional space (~300Mb)
- Map of MFG/PRO program and component relationships
- Loaded from the MFG/UTIL utility



See *Loading the Source Code Cross-Reference (Optional)* in the *MFG/PRO eB Installation Guide, Progress on Unix*

Source Code Cross-Reference

- Run MFG/UTIL from eB Installation Directory
 - From *Database* menu, select *Load Data into Database*
 - Enter *Path/HelpDB* in the Physical Database field
 - Specify *eBDir/mfghelp* as the Input Directory
 - All Files(Tables) are selected by default – Select OK
 - Complete load
- Review `mfgutil.log` for errors



See *Loading the Source Code Cross-Reference (Optional)* in the *MFG/PRO eB Installation Guide, Progress on Unix*

PROGRESS Client Installation

- Different media than the UNIX server PROGRESS
- Should be installed on a network file server,
providing bandwidth is adequate
- Installation directory must be network shareable
- Could be installed on each client
 - Requires 225-350Mb of disk space on each client
 - Requires multiple upgrades at upgrade time
- If installing on a network, make sure to follow the PROGRESS network set-up instructions

PROGRESS Client Installation

Take note of the directory for your installation of PROGRESS.

MFG/PRO for Windows Installation

- Separate media from MFG/PRO Server
 - Contains GUI object code, and optional source code
 - Contains no database build files
 - Contains GUI client setup programs
- Should be installed on a network file server
- Installation directory must be network shareable
- Installs a client version of MFG/UTIL to configure database sets, connections, and parameter files for the GUI MFG/PRO sessions

MFG/PRO for Windows Installation

Take note of your MFG/PRO for Windows installation directory.

Windows Client Setup

- Map client(s) to the networked PROGRESS and MFGPRO Client installation directories
- Run `setup.exe`, located in the MFGPRO for Windows installation directory (on file server)
- Prompts for:
 - Local installation directory (for a small number of files)
 - PROGRESS and MFG/PRO for Windows directories
 - Video resolution
 - Invokes MFG/UTIL to prompt for database set information, including networking parameters
- Settings from first install can be saved to set up subsequent clients in an Express Setup



See *Gui Client Setup* in the *MFG/PRO eB Installation Guide, Progress on Unix*

Networking Parameters

- -H (Host name)
 - Machine name where the database server is running
 - Correlates to an IP address in the Hosts file
 - May use DNS (Dynamic Name Search)
 - Client must be able to “ping” the machine name
- -S (Service name)
 - Symbolic name for a TCP/IP communication port
 - Correlates to port number in the Services file
 - Start database server with same symbolic name/port number
- -N (Network type)
 - Must be TCP/IP protocol

MFG/PRO Icons

- Client Setup creates MFG/PRO start-up icons
 - Runs a multiuser MFGPRO client session against the databases in the database set
- Example icon command line:

```
p:\dlc\bin\prowin32 -pf c:\qad\training.pf
  -ininame c:\qad\progress.svg
```

Component:	Description:
p:\dlc\bin\prowin32	PROGRESS 32-bit Windows client executable
-pf	Path and name of the parameter file containing database connection and other parameters
-ininame	Path and name of the .ini file containing the path to DLC, propath, and video resolution

MFG/PRO Icons

The icon command line is also known as the shortcut.

Sample Parameter File

```
-db mfgtrain -H UXServ1 -S train91-srv -N tcp
-db admprod -ld qadgui -H UXServ1 -S adm91-srv -N tcp
-db hlpprod -ld qadhelp -H UXServ1 -S help91-srv -N tcp
-cpinternal iso8859-1 -cpstream ibm850 -mmax 4096 -p mf.p
```

Parameter	Description
-db	Physical name of the database to connect to
-ld	Alternate logical name for the database
-H, -S, -N	Networking: Host, Service name, and Network type
-cp....	Code page definition parameters
-mmax	PROGRESS client memory management
-p	Indicates a program name to execute

Sample Parameter File

Understanding the Windows client parameter file can help you better understand client/server connections.

Sample Hosts File

```
db hosts addresses and names

192.152.71.42    UXServ1
192.152.71.43    UXServ2
192.152.71.44    UXServ3
192.152.71.45    UXServ4
```

IP Address *Host Name*

Sample Hosts File



See *Modifying Hosts and Services Files* in the *MFG/PRO eB Installation Guide, Progress on Unix*

Sample Services File

```
#
#MFGPRO Database service names and sockets

help91-srv 2511/tcp # mfghelp database
train91-srv 2521/tcp # mfgtrain database
demo91-srv 2531/tcp # mfgdemo database
adm91-srv 2541/tcp # admin database
prod91-srv 2551/tcp # production database
```

*Service
Name*

*Port No./
Protocol*

Comment

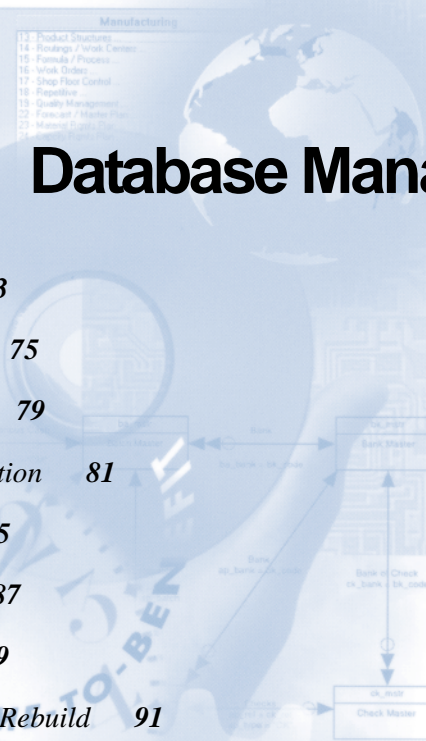
Sample Services File



See *Modifying Hosts and Services Files* in the *MFG/PRO eB Installation Guide, Progress on Unix*

Database Management

<i>Data Movement</i>	73
<i>Database Backups</i>	75
<i>Delete and Archive</i>	79
<i>Database Fragmentation</i>	81
<i>Dump and Load</i>	85
<i>Dump Procedure</i>	87
<i>Load Procedure</i>	89
<i>Bulk Load and Index Rebuild</i>	91
<i>Special Dump/Load</i>	92



Routing Maintenance (Date Based)	
Routing Code:	10-15000
Operation:	20
Standard Operation:	
Work Center:	1030 INSPECTION, ALL SITE
Machines:	
Description:	INSPEC PER PROC-00%
Machines per Op:	1
Overlap Units:	1
Queue Time:	1.0
Wait Time:	0.0
Setup Time:	0.0
Used by Product Line:	...

Data Movement

- Backup
- Delete/Archive
- Dump/Load
- Bulk Load

Data Movement

Types	When	What It Does	Comment
Backup	Regularly (day, week).	Makes exact copy of database or of changes (after image).	Restore as needed for recovery.
Archive/Delete	After data is not needed anymore.	Frees up space in database. Temporary increase in disk usage until archive/delete.	Back up before delete.
Dump/Load	Installing new MFG/PRO version or recovery of database space.	Copies data from and to database sequentially into blocks.	Time consuming. No database content affected.

Data Movement

The above table summarizes the areas of database management covered in this chapter.

Data Movement Methods

- Backup
 - UNIX utilities: cpio, dd, tar, etc.
 - PROGRESS utilities: probkup and prorest
- Delete and Archive
 - Various MFG/PRO functions
- Dump and Load
 - MFG/PRO
 - PROGRESS Data Dictionary
 - Proutil (load only)
 - MFG/UTIL

Database Backups

- You should do a full backup **every night**
 - You can perform incremental backups (transactions since last full backup), but they add complication
- Recommended method:
 - Shut down the database server (off-line backup)
 - Truncate the before-image file
 - Use any UNIX file backup utility
- As an option, you can use Probkup/Prorest
 - Only option if the server is running (on-line backup)
 - Only backs up the .db and .bi files (or extents)
- Verify your backup strategy often

Database Backups



See *Database Backups* in the *PROGRESS Systems Administration Reference, Progress Version 9*

Backup Utilities

UTILITY	TO	WHAT	TYPE
UNIX cpio	Tape	All Files	Incremental or Full
UNIX tar	Tape or Disk	All Files	Incremental or Full
PROGRESS probkup	Tape or Disk	db & bi extents only	Incremental, Full or online*

* When using probkup -online option: database server and after-imaging may be running. For full or incremental backups, AI must be stopped prior to the backup.

Backup Preparation

DO	USE	WHY
Shut down the server	stop. <i>SetName</i> –or– proshut = <i>_mprshut</i>	Avoid data corruption. PROGRESS backup using -online option avoids corruption, but is only 99.9% accurate.
Truncate your .bi file	proutil	This eliminates data from the bi extents, but saves the file.

Full vs Incremental Backups

- **Full Backups Only (Recommended)**
 - Full database backups daily
 - Restore: the most recent (good) full backup
- **Full and Incremental Backups (Cumulative)**
 - Full database backup weekly (Friday) and incremental backups daily for all days since last full backup
 - Example: on Monday, back up Monday changes, and on Tuesday, back up Monday and Tuesday changes
 - Restore: the most recent full backup followed by the most recent incremental backup
- **Full and Incremental Backups (Non-Cumulative)**
 - Full database backup weekly (Friday) and incremental backups daily for each day's changes
 - Example: on Monday, back up Monday changes, and on Tuesday, back up only Tuesday changes
 - Restore: the most recent full backup followed by each of the incremental backups in sequence

Delete and Archive

- MFG/PRO includes delete/archive functionality throughout the system
 - You should delete and archive records within given time periods.
- Delete/Archive functions use date, and other criteria to identify which records to delete
- Result:
 - Deletes the data from the database
 - Produces a flat ASCII history file containing the raw data that was deleted

Delete and Archive

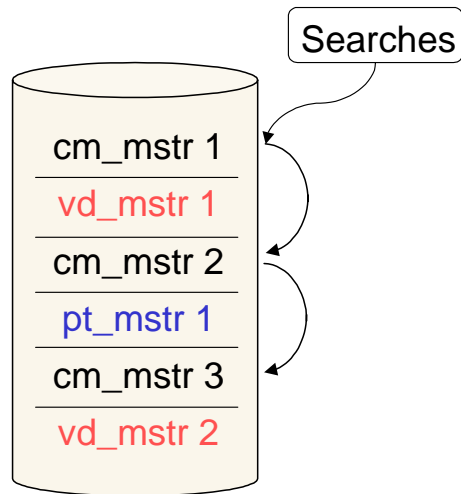
For a full explanation of delete and archive functions, refer to the *Manager Functions Training Guide*.

Delete and Archive (cont.)

- Archive/Delete should be run to trim excess data to specific history retention period
 - Contact users to determine what to delete
- Always run delete/archives in report mode first to verify proper deletion
 - For a report, set Delete and Archive flags to No
- You can reload data into the database using Archive File Reload
- Mass deletes cause database fragmentation

Database Fragmentation

- A database has many different logical files within it
 - As records are added, the files become fragmented and no longer contiguous
 - The files each contain different types of data (parts, vendors, etc.)
- Fragmentation causes inefficient logical file searches
- Fragmentation is increased by delete/archive because the location of deleted data is reused for non-similar data



Database Fragmentation



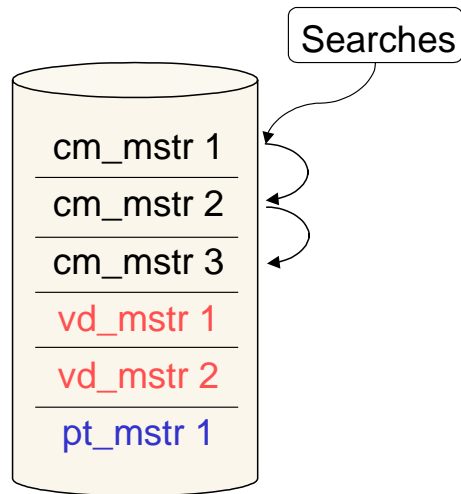
See *Database Dump and Load* in the *PROGRESS Systems Administration Reference, Progress Version 9*

Use Dump/Load to Defragment

- By dumping a fragmented database, the contents of the noncontiguous logical files are replicated to flat ASCII data files (.d)
 - cm_mstr.d file:
"data1", "data2", "data3", "data4"
"data1", "data2", "data3", "data4"
"data1", "data2", "data3", "data4"
 - vd_mstr.d file:
"data1", "data2", "data3", "data4"
"data1", "data2", "data3", "data4"
 - pt_mstr.d file:
"data1", "data2", "data3", "data4"

Use Dump/Load to Defragment

- You load the dumped data files into a copy of MFG/EMPTY
- As the data is loaded, the target database grows in size to accommodate the data
- The flat ASCII files contain similar records, so they are loaded together with all the cm_mstr data, then the vd_mstr data, etc.
- The reloaded database is as contiguous as it can be



Dump/Load Benefits

Reasons for dumping and loading data:

- Reclaim disk space
- Maintain database integrity after delete/archive
- Reduce disk fragmentation

Dump and Load

- Dump replicates database data into .d files
 - File name is the logical file name with extension .d
 - Dump does not delete any data from the database
- After a dump, you can load the .d files into an empty database that has a schema but no data (such as a copy of mfgempty.db)
- During a load, the empty database grows to the required size
 - All files load contiguously

Dump and Load



See *Database Dump and Load* in the *PROGRESS Systems Administration Reference, Progress Version 9*

Types of Dumps and Loads

- There are four methods to dump a database
 - MFG/PRO Dump/Load program (36.16.4)
 - PROGRESS Data Administration Dump
 - MFG/UTIL
 - Proutil Dump (Binary Dump – Single file at a time)
- There are five methods to load a database
 - MFG/PRO Dump/Load program (36.16.4)
 - PROGRESS Data Administration Load
 - MFG/UTIL
 - PROUTIL Load (Binary Load – Loads a binary dump file)
 - PROUTIL bulk load
 - Fastest due to index deactivation
 - Either PROUTIL option requires separate index build
 - Requires the creation of a file descriptor file (.fd file)

Dump Procedure

- 1 Check primary indexes
 - Proutil idxanal
 - To estimate the total dump time, you can identify files that represent 5 – 10% of the data, dump them, and time the process
- 2 Create a dump directory and check for adequate disk space
 - Requires 75% of the database size in free space

Dump Procedure



See *Database Dump and Load* in the *PROGRESS Systems Administration Reference, Progress Version 9*

Dump Procedure (cont.)

3 Verify you are the only user on the system

- Single-user shared memory connection at the database server
- Multi-user network connection as only one logged on
 - Move mf.p to prevent other from logging on
 - Network connection not recommended for large dump or load

4 Choose one of the following methods:

- In MFG/PRO, run Database File Dump/Load
- Use the PROGRESS Data Dictionary dump
- Use MFG/UTIL



Complete the following Lab: “Dumping Through PROGRESS Data Administration Tool.”

Load Procedure

- 1 Prepare a copy of the empty MFG/PRO database (mfgempty.db)
 - Example: `$DLC/prodb NewProdDb.db mfgempty`
- 2 If you want to bulk load, create a description file
 - In the Data Dictionary, Admin menu, choose Create Bulk Loader Description File
- 3 Verify you are the only user on the system
 - Single-user shared memory connection at the database server
 - Multiuser network connection as only one logged on
 - Move mf.p to prevent others from logging on
 - Network connection not recommended for large dump or load

Load Procedure



See *Database Dump and Load* in the *PROGRESS Systems Administration Reference, Progress Version 9*

Load Procedure (cont.)

4 Choose one of the following methods:

- Run the Bulk Loader option of the PROGRESS PROUTIL utility (recommended)

```
proutil DbName -C bulkload DescriptionFile
```

- Use the PROGRESS Data Dictionary load
 - In the Data Dictionary, Admin menu, choose Load Data and Definitions, then Table Contents
- In MFG/PRO, run Database File Dump/Load (36.16.3)
- Use MFG/UTIL

5 If you used the bulkloader, rebuild the indexes

- Command: `proutil DbName -C idxbuild`
- **NOTE:** In the event of a load failure, start over at the beginning



Complete the following Labs: “Creating and Using a Copy of mfgempty,” and “Loading Through PROGRESS Data Administration Tool.”

Bulk Load and Index Rebuild

- After a bulk load, rebuild database indexes
 - You may also need to rebuild for system maintenance
- Use the idxbuild option of the PROUTIL utility
- The rebuild is quicker if disk space is available for index sorting
 - Requires 75% of the database size
 - Use the -TB and -TM parameters to speed the sort
- If the resulting sort file will exceed the maximum file size limit, use a multi-volume sort file
 - Create a file to indicate the sort file extents
 - Stipulate this file as part of the PROUTIL command

Bulk Load and Index Rebuild



See *Proutil - Bulk Load* in the *PROGRESS Systems Administration Reference, Progress Version 9*



Complete the following Labs: “Creating the Bulk Loader Description File,” “Bulk Load the Data,” and “Rebuilding the Indexes.”

Database Load with Default Data

- MFG/PRO eB media contains special data directories – Mfgtrain, Mfgdemo, Mfg, Admin, etc
- These directories contain non language specific files only. Language specific files are located in *eBDir/LangDir/Mfg* and *eBDir/LangDir/Admin*
- When loading a database with the data from a MFG/PRO eB directory, you must compliment the contents with the appropriate *Language Files*
- Loading a Database with this default data requires 2 loads, one from the data directory, and one for the specific language directory

Database Load with Default Data

- The Main MFG/PRO eB Database and the Administration Database have language specific components.
- The language specific files in the Main Database:
 - mnt_det (Menu Detail)
 - msg_mstr (Message Master)
 - lngd_det (Language Detail)
- The language specific files in the Admin Database:
 - mnts_det (Translation Detail)
 - lbl_mstr (Label Master)

Special Dump/Load

- Multi-Stream Dump/Loads
 - To speed the process, you can run several sessions, with each dumping or loading a portion of the files
 - This is only beneficial if each session is writing to or reading from a separate I/O channel – and even then the benefit is minimal
- Limited Disk Space
 - You must have enough space to dump the largest file (+10%), as seen through the file size report
 - Dump enough files to fill 90% of the available space
 - Back up to tape, delete from disk, and repeat procedure

Special Dump/Load



See *Limited Disk Space Dump and Load* in the *PROGRESS Systems Administration Reference, Progress Version 9*

Dump/Load Multi-Stream DB

- 1 Start a server session
- 2 Run MFG/PRO in three separate sessions
- 3 Run 36.16.3 for dump `aa` to `lz` on session 1, `ma` to `rz` on session 2, and `sa` to `zz` on session 3
- 4 Back up old database

Dump/Load - Limited Space

- 1 Start MFG/PRO
- 2 Check available disk space
- 3 Try dumping the tr_hist (or largest) file; create free disk space as needed
- 4 Check disk space and byte size of tr_hist
- 5 Run Database File Size Report and correlate record count with bytes or use Proutil's db analysis option
- 6 Subtract 10% from available free space
- 7 Estimate range of files to dump that will fit in space
- 8 Run Dump utility on appropriate file range
- 9 Move the .d files to tape or diskette
- 10 Do steps 7 through 9 until all files are dumped



See *Limited Disk Space Dump and Load* in the *PROGRESS Systems Administration Reference, Progress Version 9*

Performance Tuning

Tuning Techniques 97

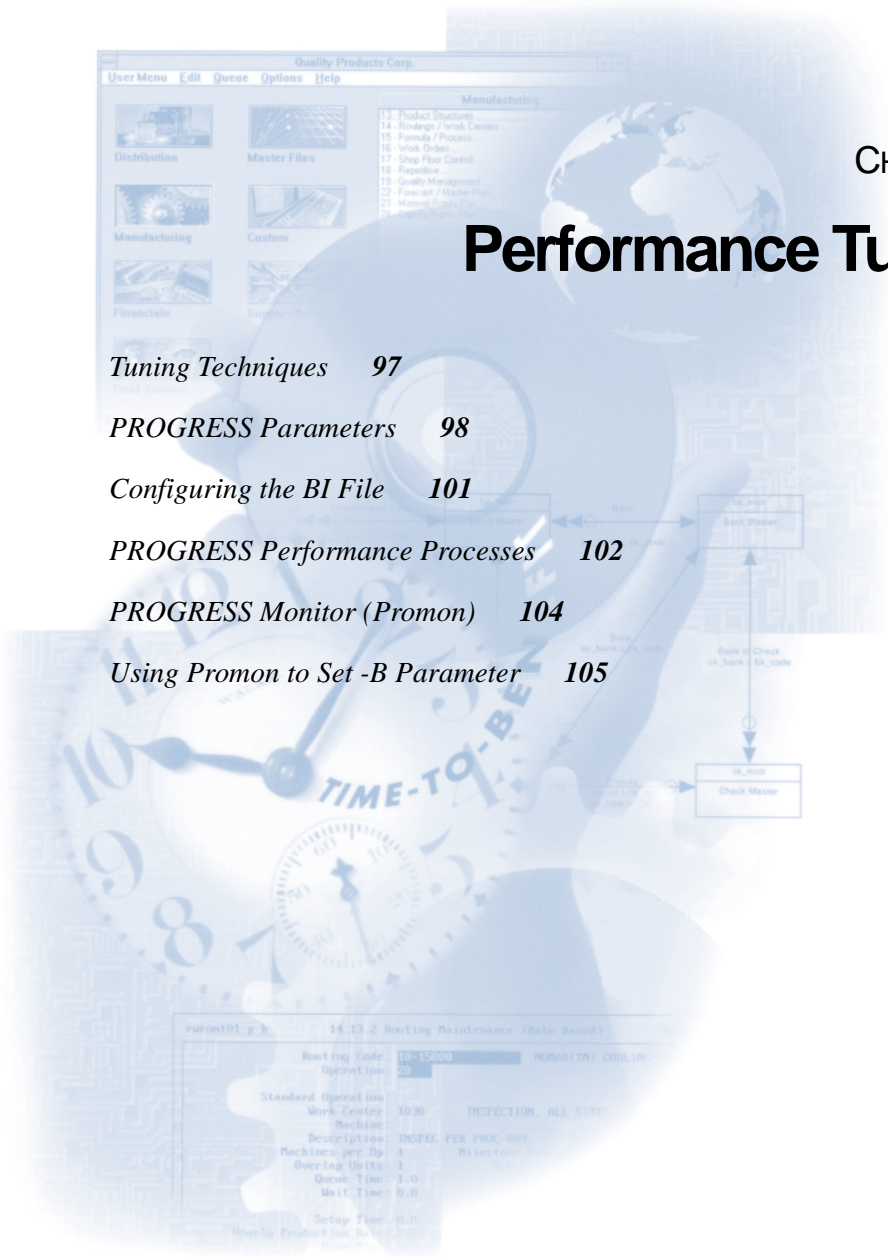
PROGRESS Parameters 98

Configuring the BI File 101

PROGRESS Performance Processes 102

PROGRESS Monitor (Promon) 104

Using Promon to Set -B Parameter 105



Performance Tuning

- Progress Parameters
- Primary Recovery Area Configuration
- BIW, AIW, APW, WatchDog
- PROMON

Tuning Techniques

- Use PROGRESS parameters to utilize resources efficiently
 - Examples: -B, -mmax, -spin
- Reduce I/O contention
 - Relocate the before-image extents to another I/O channel
 - Specify a separate location for the before-image extents within the multi-volume structure
 - Supply multiple I/O channels into the physical database
 - Specify multiple locations for the data extents within the multi-volume structure
- Utilize PROGRESS Performance Aids
 - BIW, AIW, APW, and Watchdog

Tuning Techniques

The above techniques are explained further in this chapter.

PROGRESS Parameters

- PROGRESS executables have arrays of parameters that can be specified to enhance performance
- These parameters must be passed to the PROGRESS executable directly
- Specific parameters are applicable to specific executables
- Certain parameters must be tuned over time, using the PROGRESS Monitor to evaluate the effects on the database

PROGRESS Parameters

Table 4.1
Performance Parameters

Option	Effect	Where To Set	Notes
-1	Single User	End User (<code>prowin32</code>)	Identifies the database is connected without using a server in a single-user mode. Not an option for client/server connections.
-B	Database Buffers	Server Start (<code>_mprosrv</code>)	Allocates a specified amount of system memory into which you can read and manipulate database records. It is much faster to manipulate a record in memory rather than through the I/O channel. This is a critical setting for system performance, and is set while using the PROGRESS Monitor.

Option	Effect	Where To Set	Notes
-bifufs	BIW	Server Start (_mprosrv)	Increases the number of before-image buffers in the before-image buffer pool. Default is 5. Initial value recommendation is 20. Increase if there are empty buffer waits in the Promon activity.
-aibufs	AIW	Server Start (_mprosrv)	Increases the number of after-image buffers in the before-image buffer pool. Default is 1. Initial setting recommendation is 5 or 1.5 times the value of the -bibufs parameter.
-Mf	BI	Server Start (_mprosrv)	Delays PROGRESS from synchronously writing out to disk the last BI file record at the end of each transaction. Default is 3 milliseconds. Use to improve performance of heavily loaded shared memory systems. At system failure, the value determines the oldest complete transaction that can be lost.
-spin	Resource Waits	Server Start (_mprosrv)	The number of times a process retries to acquire a latch on shared memory data structures. Default is 0. If the value is greater than 0, then a spin-lock algorithm is used which is more efficient than using semaphores when you have multiple processors. Initial value recommendation is 2,000. If this consumes too much CPU, decrease. Use the Promon R&D Performance Indicators report to view resource waits.
-Ma	Remote Users	Server Start (_mprosrv)	The maximum number of remote users per DB server. Default is the maximum number of users (-n param) divided by the Max servers parameter (Mn). Default is 5 users per server. If a given number of remote clients overloads the server, or depletes the file descriptors on the system, set Ma to limit the clients per server.
-Mn	Remote Users	Server Start (_mprosrv)	The maximum number of remote client servers that can be started by the broker process. Default is 4. Performance tradeoff to consider: swapping overhead for many servers vs. overloading a server with too many clients.
-Mi	Remote Users	Server Start (_mprosrv)	Minimum number of remote clients on a server before the broker starts another the broker process. Default is 1. If you significantly increase Mn, you should also increase Mi. For example, if Mn was set to 10 with no Mi increase, 10 servers for the first 10 remote users would run, which may be fine if system resources support it.
-L	Lock Table	Server Start (_mprosrv)	Specifies how many records from the database can be locked at any one time cumulatively between all the users. May be useful for CIM load.
-n	# of Users	Server Start (_mprosrv)	Specifies how many users can access the database at any given time. The system adds one to this number and reserves it for the PROGRESS Monitor. Typically, at the start of a MFG/PRO session, the user connects to several databases. The lowest -n inhibits the users from connecting.

Option	Effect	Where To Set	Notes
-g	Before-Image File Name	Everywhere	Used by <code>_mprosrv</code> if the database is accessed using a server. It is passed to <code>_progres</code> if needed during single-user access, as well as to <code>PROUTIL</code> or other <code>PROGRESS</code> tasks connecting single users. Its purpose is to direct the system to the location of the <code>.bi</code> file if it has been moved anywhere other than the database directory. NOTE: <code>PROGRESS</code> recommends not using this parameter and instead relocating the <code>.bi</code> file through a multi-volume structure.
-mmax	Maximum Memory	Client Start (<code>prowin32</code>)	The initial amount of memory allocated for r-code segment in Kb. <code>PROGRESS</code> dynamically allocates space for r-code in the execution buffer as needed. Once <code>mmax</code> is reached, <code>PROGRESS</code> swaps non-active segments to the sort file. This is a soft-limit so that if more memory is required, it dynamically allocates it. Default is 512.
-D	Directory Entries	Client Start (<code>prowin32</code>)	Specifies the length of the user's directory tables; i.e., the number of files a user can have open at any given point.
-a	After-Image File Location	Everywhere	Same as <code>-g</code> , but pertains to the <code>.ai</code> file, which is optional. If you use roll forward recovery, the server, or the user's session, needs to know where the <code>.ai</code> file is. This specifies where to find the file.
-T	Temp Files	Client Start (<code>prowin32</code>)	<code>PROGRESS</code> creates temp files, and sort files are created in the directory <code>MFG/PRO</code> was started from. You can redirect these files by using the <code>-T</code> followed by the directory.
-TB	Temp File Block Size	Client Start (<code>prowin32</code>)	<code>PROGRESS</code> allocates a certain size to temporary files. When this size is exceeded, system operation halts, and then reallocates a new size. By using <code>-TB</code> , you can specify a larger block size for temporary files so that the system does not have to stop and reallocate as often. This parameter is particularly helpful during an index build.
-s	Stack	Client Start (<code>prowin32</code>)	The size of the stack in 1k units. Default is 40. Increase if you receive stack overflow messages (usually during <code>.df</code> loads on large tables), or during recursive procedures.
-stsh	Stash	Client Start (<code>prowin32</code>)	The size of the stash in 1k units. Default is 2. Stash area is temporary storage for modified index fields. Calculate by taking the size in bytes of the largest field with a word index, double the value and add 1k. <code>PROGRESS</code> terminates with a fatal error if it runs out of stash.
-TM	Sort Speed	Client Start (<code>prowin32</code>)	Increases the speed at which the <code>idxbuild</code> sort is performed.
-by	Bypass	Server Shut (<code>_mprshut</code>)	When the database is shut down, you are given a four-selection menu. Using <code>-by</code> bypasses this menu for unattended shutdown operations.

Configuring the BI File

- The bi extents undergo heavy I/O
 - PROGRESS writes to them during most transactions
- By default, both the database and bi extents reside on the same physical location, creating severe I/O contention
- Moving the bi extents to a separate I/O channel eliminates some I/O contention
- To relocate them create a multi-volume structure and locate the bi extents on a separate I/O channel

Configuring the BI File



See *The Primary Recovery Area* in the *PROGRESS System Administration Reference, Progress Version 9*

PROGRESS Performance Processes

- Before Image Writer (BIW)
 - Run one per database
 - Buffers a block of .bi file writes, and commits the entire block at once
 - To start: `DLC/bin/probiw dbname`
- After Image Writer (AIW)
 - Run one per database
 - Buffers a block of .ai file writes, and commits the entire block at once
 - To start: `DLC/bin/proaiw dbname`

PROGRESS Performance Processes



See *PROGRESS Performance Aids* in the *PROGRESS System Administration Reference, Progress Version 9*

PROGRESS Performance Processes

- Asynchronous Page Writer (APW)
 - Run one or two per database
 - Buffers a page of .db file writes, and commits the entire page at once
 - To start: `DLC/bin/proapw dbname`
- Watchdog (Prowdog)
 - Run one per database
 - Identifies and eliminates “orphaned” processes after a set time
 - To start: `DLC/bin/prowdog dbname`

PROGRESS Monitor (Promon)

- Promon is a database monitoring utility
 - To start: `DLC/bin/promon dbname`
- Supplies details about:
 - Users Statistics
 - Locking and Waiting Statistics
 - Block Access
 - Record Locking
 - Activity
 - Shared Resources
 - Database Status
- Useful for determining -B setting and evaluating database performance

PROGRESS Monitor (Promon)

Useful PROMON Screens

Menu 3, Selection 1

Menu 5, Activity Server

Menu 6, Shared Resources

Menu 7, DB Status



See *PROMON – PROGRESS Monitor* in the *PROGRESS System Administration Reference, Progress Version 9*

Using Promon to Set -B Parameter

- Promon option 3 shows database reads & requests
- Promon option 5 shows the percent of Buffer Hits
 - Buffer Hit is the percent of database data read from fast memory buffers
 - You should target 90% Buffer Hits
- Use to determine -B setting
 - Set -B parameter in the database server start-up script
 - Use promon to analyze Buffer Hits during peak usage
 - Increase -B setting and restart server
 - Repeat procedure until Buffer Hits drop off, and reset closer to previous setting; until you reach the optimal value

Using Promon to Set -B Parameter



See *Block Access (PROMON Item 3)* in the *PROGRESS System Administration Reference, Progress Version 9*

Promon Examples

Promon Item 3: Block Access

Type	Usr	Name	DB Reqst	DB Read	DB Writes	BI Reads	BI Writes	AI Reads	AI Writes
ACC	999	TOTAL	804590	72152	58	78	477	0	0
ACC	0	root	257	25	2	59	458	0	0
ACC	1	sjtrain1	4794	1132	0	0	0	0	0

Promon Item 5: Activity (excerpt)

```

Rec Lock Waits    0 %   BI Buf Waits 0 %   AI Buf Waits 0 %
Writes by APW    0 %   Writes by BIW 0 %   Writes by AIW 0 %
Buffer Hits      98 %
DB Size          25536 K   BI Size 2688 K   AI Size 0 K
FR chain         189 blocks  RM chain 76 blocks
Shared Memory 33683 K   Segnebts 3

0 Servers, 0 Users <0 Local, 0 Remote, 0 Batch>,0 Apws
    
```

Multi-Volume Database

<i>Multi-Volume Overview</i>	109
<i>Multi-Volume Components</i>	110
<i>Multi-Volume Creation Steps</i>	111
<i>Creating a Structure File</i>	112
<i>Sample Structure File</i>	113
<i>Determining Database and Extent Size</i>	114
<i>Calculating the Structure File Extent Size Number</i>	115
<i>Using Prostruct</i>	116
<i>Creating a Void Multi-Volume Database</i>	117
<i>Populating a Void Multi-Volume Database</i>	118
<i>Copying a Single Volume to Multi-Volume</i>	119

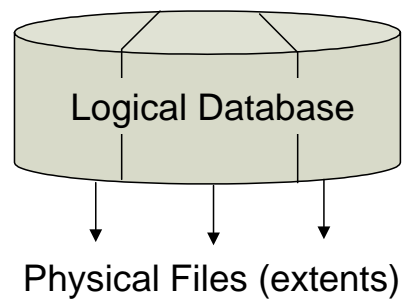
Routing Maintenance (Date Based)	
Routing Code	10-15008
Operation	20
Standard Operation	1030
Work Center	INSPECTION, ALL SITE
Machines	1
Description	INSPEC PER PROC 60%
Machines per Sp	1
Overlap Units	1
Queue Time	1.0
Wait Time	0.0
Setup Time	0.0

Multi-Volume Database

- Components
- Creation
- Loading data

Multi-Volume Overview

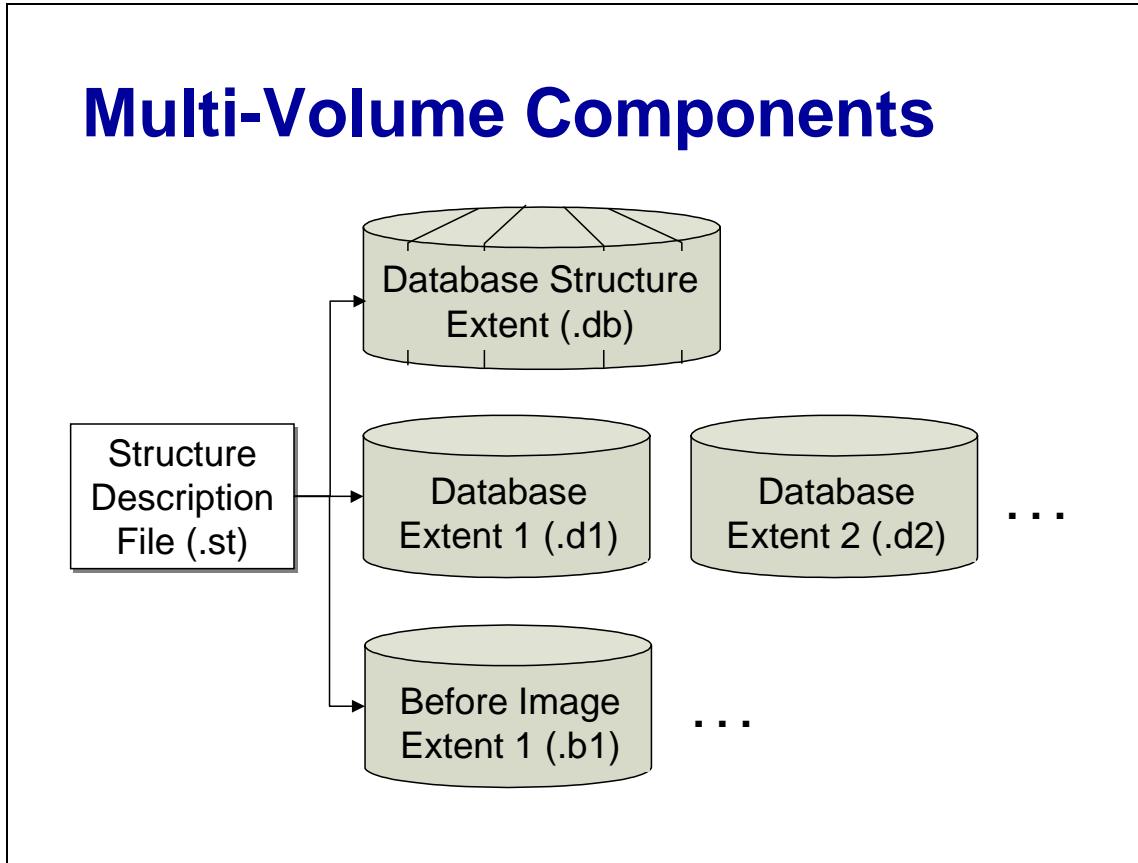
- One logical PROGRESS V9 database is comprised of several physical files
- These physical files can be placed across several physical disk drives to maximize I/O ability
- The before-image and after-image extents can also be made up of many physical files and located on specific I/O channels



Multi-Volume Overview



See *Progress Database Architecture* in the *PROGRESS System Administration Reference, Progress Version 9*



Multi-Volume Components

 See *Database Components PROGRESS System Administration Reference, Progress Version 9*

Multi-Volume Creation Steps

- Create a structure file
 - Specifies the type of extents: data, bi, or ai
 - Specifies the Data Storage Area Name and Number
 - Specifies the number of records per block
 - Specifies the directory location of the extents
 - Specifies the extent sizes
- Use prostrct to create or add the described extents
- Add schema and or data to this *void* structure
 - Use procopy (If Target has all Data Areas of Source)
 - Can copy both schema and data in one step
 - Or, use any load method to load schema or data

Multi-Volume Creation Steps



See *Database Creation* in the *PROGRESS System Administration Reference, Progress Version 9*

Creating a Structure File

A typical line in a structure file:

```
d "TRANSACTION":7,64 /usr/data1 f 5120
```

Component	Description
d	Indicates this extent will contain data; could also be "a" (ai), or "b" (bi)
Transaction:7	Specifies the name:number of the Data Storage Area this extent is part of
64	The number of data records of this type that fill a system I/O block
/usr/data1	Directory Location(I/O channel) that the described extent will reside upon
f	Indicates this extent is to be fixed in size. No "f" means extent is variable, and will grow when/as needed
5120	Extent size, express in 1024 byte blocks. Must be a multiple of 16

Creating a Structure File



See *Creating the Structure Description File* in the *PROGRESS System Administration Reference, Progress Version 9*

Sample Structure File

b /db f 192	Allocate 10-20% database size in fixed bi extents
b /db	AREA 3 – required – only needs locations
#	
a /ai f 64	Allocate several small files to contain the heaviest expected daily transaction rate data (or time between backups). Not required to specify Data Storage Area Name: Number or Records/block.
a /ai f 64	
a /ai	
#	
d "Schema Area":6,32 /db	AREA 6 – “Schema Area”
#	
d "TRANSACTION":7,64 /user/data1 f 5120	User Defined Area 7 dedicated to transaction data
d "TRANSACTION":7,64 /user/data1 f 5120	User Defined Area 7 has two fixed extents
d "TRANSACTION":7,64 /user/data1	**64 rec/blk, specific I/O channel, variable extent
#	
d "TRANSACTION_IDX":8,32 /user/data2 f 5120	User Defined Area 8 holds transaction data indexes
d "TRANSACTION_IDX":8,32 /user/data2	**Could have many extents & or many I/O channels
#	
d "STATIC":9,64 /user/data3 f 5120	User Defined Area 9 holds static data
d "STATIC":9,64 /user/data3	**Static data seldom changes = little I/O
#	
d "STATIC_IDX":10,32 /user/data3 f 5120	User Defined Area 10 holds the static data indexes
d "STATIC_IDX":10,32 /user/data3	**Grouped with the actual data since little I/O

Sample Structure File



See *Example Structure File* in the *PROGRESS System Administration Reference, Progress Version 9*

Determining Database and Extent Size

- Total database size factors
 - Size of existing database (if converting an existing database)
 - Expected growth
 - Target a 2 – 4 Gb database for a new system
 - Default Eb database is 60+ extents, and 30+ Data Storage Areas
- Extent size factors
 - Determine range of data amounts in each Data Storage Area
 - Use Proutil dbanal, plus MFG/PRO Database Storage Areas appendix
 - Allow for expected growth in each area
 - Typical extent sizes range between 25MB – 100MB due to number of extents in eB database structure
 - Ideal extent size is 250MB, but do not exceed 500Mb extents
 - Create enough extent capacity in each area for expected use, and minimize lesser used areas to achieve desired total size

Determining Database and Extent Size



See *Determining the Allocation Sizes* in the *PROGRESS System Administration Reference, Progress Version 9*.

Calculating the Structure File Extent Size Number

- Once you know the total database size and expected data usage of each data type, you can determine the number and size of extents
- The Extent Size Number must be a multiple of 16 and expressed as a 1024-byte block

Step	Example
1 Start with the extent size	100MB = 100,000,000 bytes
2 Divide by 1024 (bytes in each block)	$100,000,000 \div 1024 = 97656.25$
3 Divide by 16	$97656.26 \div 16 = 6103.515625$
4 Round up if necessary	6104
5 Multiply by 16 and use as the block size	$6104 \times 16 = 97664$ $(97664 \times 1024 = \text{approx. } 100\text{MB})$

Calculating the Structure File Extent Size Number



See *Converting the Extent Size to Blocks* in the *PROGRESS System Administration Reference, Progress Version 9*

Using Prostrct

- Prostrct is the multi-volume database utility
- Prostrct has many options
 - Create: generates a new, void multi-volume structure
 - Add: adds extents to an existing structure
 - Requires an additional structure file specifying just new extents
 - The previous variable extent becomes a fixed extent
 - List: creates a .st file for an existing multi-volume database
 - Statistics: shows database size and how much is used
 - Unlock: used during repair/recovery operations
- Prostrct command syntax example:

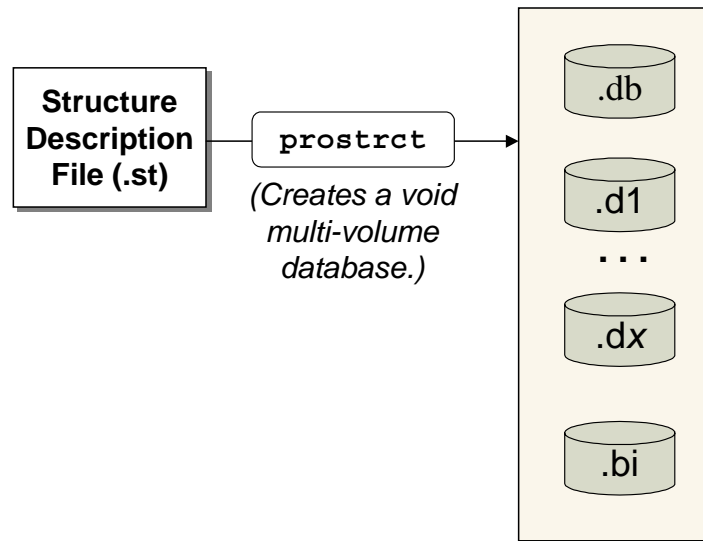
```
$DLC/bin/prostrct create /dbdir/dbname /stdir/dbname.st
```

Using Prostrct



See *Creating a Void Multi-Volume Database* in the *PROGRESS System Administration Reference, Progress Version 9*

Creating a Void Multi-Volume Database



Creating a Void Multi-Volume Database

- The prostrct command uses your structure description (.st) file to create all the extent files, as well as a .db file.
- This .db file contains no data, schema, or indexes. It contains a map to the data, bi, and ai extents.
- All commands must reference the .db file, without considering the .bi file location, what extent contains the data, etc.

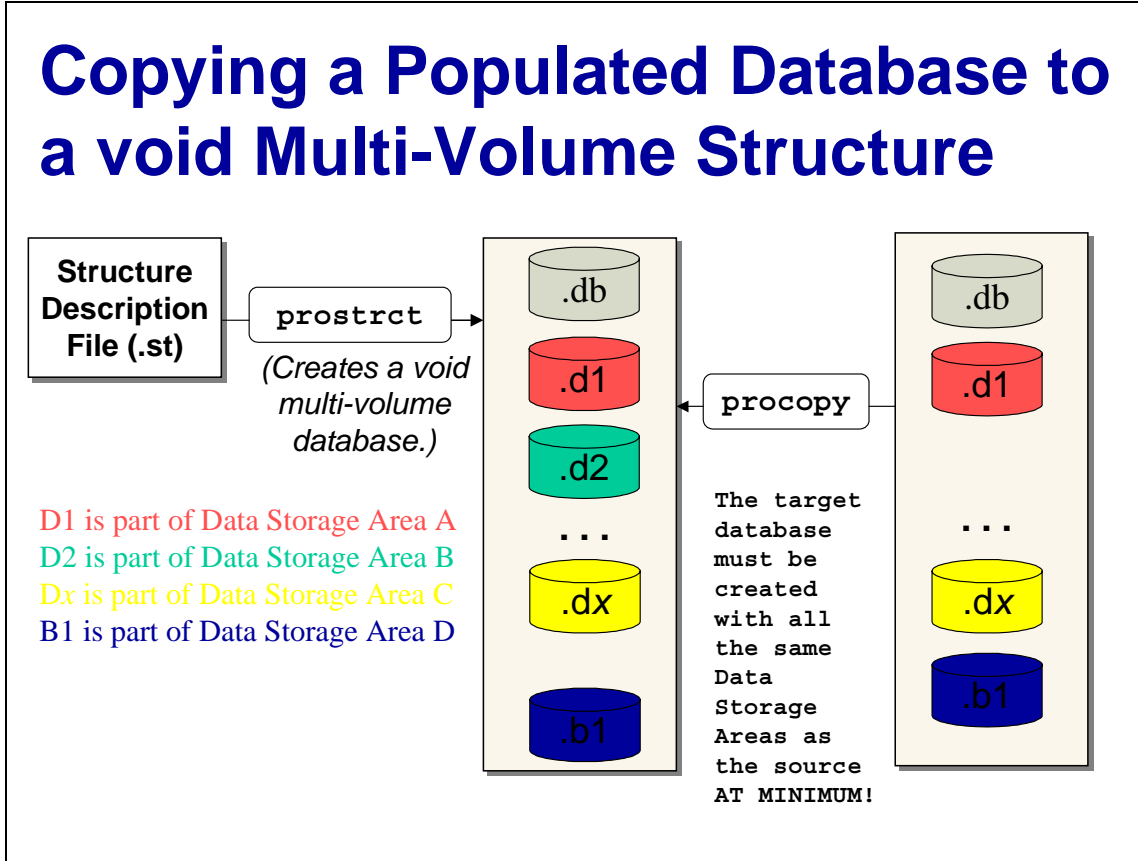
Populating a Void Multi-Volume Database

- The newly created multi-volume structure is *void*
- Schema must be loaded by either method:
 - PROGRESS Dictionary Database Definition Load
 - Procopy a database with schema to the void database
- Data can be loaded after schema is created using any standard data load method:
 - Proutil, PROGRESS Dictionary, MFG/UTIL, or MFG/PRO
- Schema and data can be loaded together
 - Procopy a populated database to the void database
 - Syntax: `DLC/bin/procopy /sourcedb /targetdb`

Populating a Void Multi-Volume Database



See *Load Schema and Data* and *Procopy* in the *PROGRESS System Administration Reference, Progress Version 9*



Copying a Populated Database to a Void Multi-Volume Structure

Copying a populated database to a void multi-volume structure is a common way to complete the multi-volume database setup. As shown in the above diagram, it is the final stage after creating a structure description (.st) file and running the prostrct command.



Complete the following Lab: “Setting Up a Multi-Volume Database.”

System Utilities

Database Set Administration 129

Compiling Programs 130

PROGRESS Batch Job 133

Using UNIX cron 134

Establishing a Batch Mode Client Session in cron 136

PROGRESS Batch Program 137

Batch Input File 138

```

parent01 p > 14.13.2 Routing Maintenance (Date Based)
-----
Routing Code: 10-15000  NONPAR (TR) COIL INS.
Operation: 20
-----
Standard Operation:
Work Center: 1030  INSPECTION, ALL SITE
Machine:
Description: INSPEC PER PROC 00%
Machines per Sp: 1  Reflection:
Overlap Units: 1
Queue Time: 1.0
Wait Time: 0.0
Setup Time: 0.0
-----
View by Product Line &
  
```

Administration Utilities

- MFG/UTIL
- Cron
- Batch

Database Set Administration

- To alter the MFG/PRO start-up, shut-down, and client connection scripts and icons, you can use MFG/UTIL
- There are separate server and client MFG/UTIL utilities
 - Server MFG/UTIL controls database maintenance, database startup files, and server code compilation
 - Server MFG/UTIL has a client section to control the Telnet Character session startup scripts and database connections
- There is a separate client MFG/UTIL to handle the Gui client connections, and Windows code compilation
- Scripts, Icons and parameter file contents are based on the database set definitions

Database Set Administration



See *DataBase Sets* and *MFG/UTIL* in the *MFG/PRO eB Installation Guide, Progress on Unix*



Complete the following Lab: “Modifying a Database Set and Startup Icons.”

Compiling Programs

- Compile character and Windows code separately using separate MFG/UTILs
 - Specify source file directory or, if used, a work file
 - Specify destination directory for saved .r code if wanted
 - Use the compile database set to ensure that you compile with the qaddb logical database name
 - Set Propath to include source and destination directory
- You can also compile via the PROGRESS Editor
- A resulting object code file is produced
 - Same file name, but with a .r extension
 - By default, MFG/UTIL saves each program under a language code directory, then a two-letter sub-directory named after the first two letters in the program name

Compiling Programs

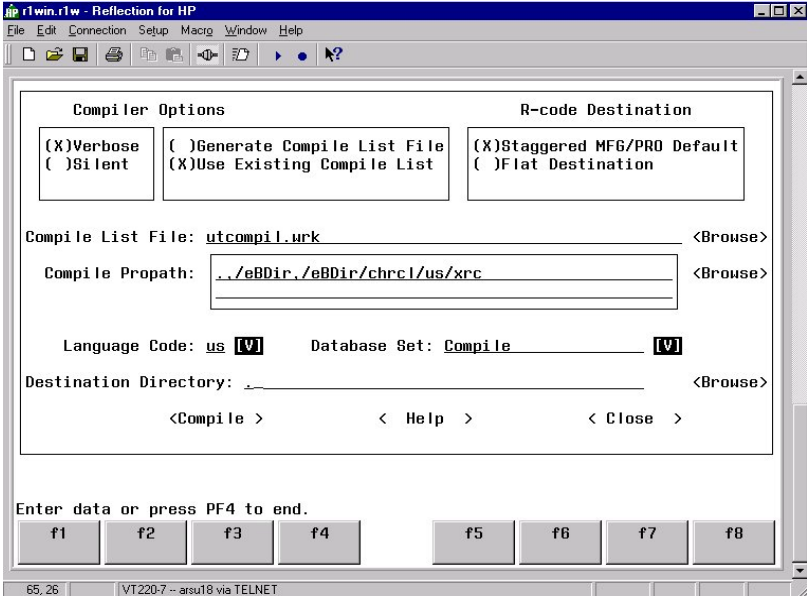


See *Compiling Programs* in the *MFG/PRO eB Installation Guide, Progress on Unix*

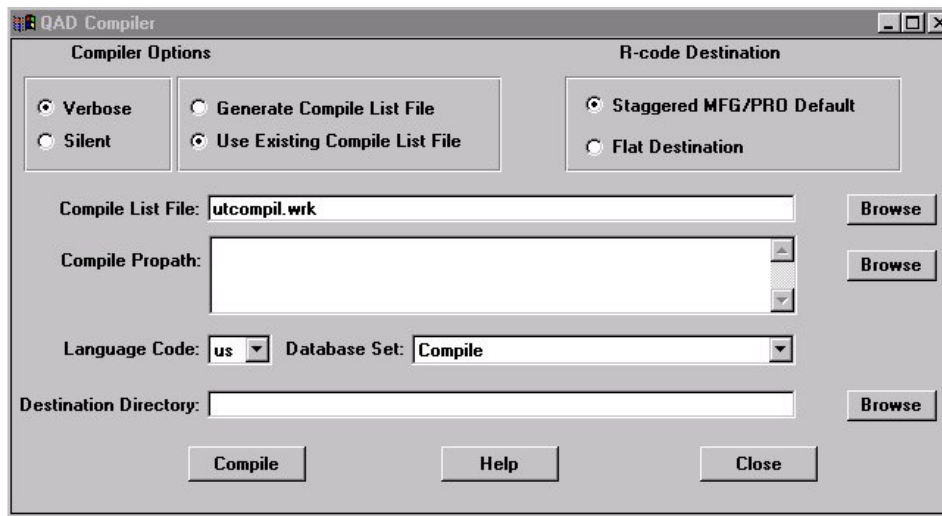


Complete the following Lab: “Compiling a Program.”

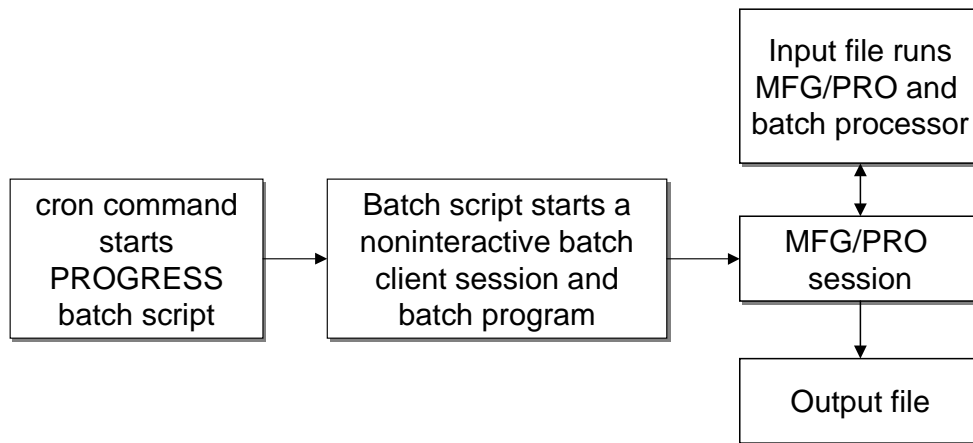
MFG/UTIL Compile Screen



GUI Client MFG/UTIL Compile Screen



PROGRESS Batch Job



PROGRESS Batch Job

The overall steps for setting up an automated batch job are:

- Establish a non-interactive, batch-mode client database connection.
- Execute a progress program.
- Specify a file to read input from.
- Specify a file to write output to.
- Invoke MFG/PRO.
- Input file directs a MFG/PRO function.



See *CRON* in the documentation for your specific Operating System

Using UNIX cron

- cron is a timer utility
- Users' cron entry files in /usr/spool/cron/crontabs
- Used to issue commands at predetermined times
- Can be used to invoke a PROGRESS batch job

Using UNIX cron



See *About the UNIX cron Utility* in the *MFG/PRO eB Installation Guide, Progress on Unix*

cron Instruction File

Minute (0-59)	Hour (0-23)	Day of Mo. (1-31)	Month (1-12)	Day of Week 0=Sunday	Script
30	23	*	*	*	/usr/scripts/backup
00	03	31	*	*	/usr/scripts/mo.end
15	22	*	*	1-5	/usr/scripts/daybatch
48	01	*	*	1,3,5	/usr/scripts/mwf

Establishing a Batch Mode Client Session

- Execute the `$DLC/bin/_progres` executable
- Pass the `-b` and other applicable parameters
- Specify the database(s) to connect to
- Specify a program to execute

```
$DLC/bin/_progres /DBPath/DBName -db  
/GuiDBPath/GuiDBName -ld qadgui -b -cprcodein  
ISO8859-1 -p batrun.p
```

Establishing a Batch Mode Client Session



See *Creating the PROGRESS Program* in the *PROGRESS System Administration Reference, Progress Version 9*

PROGRESS Batch Program

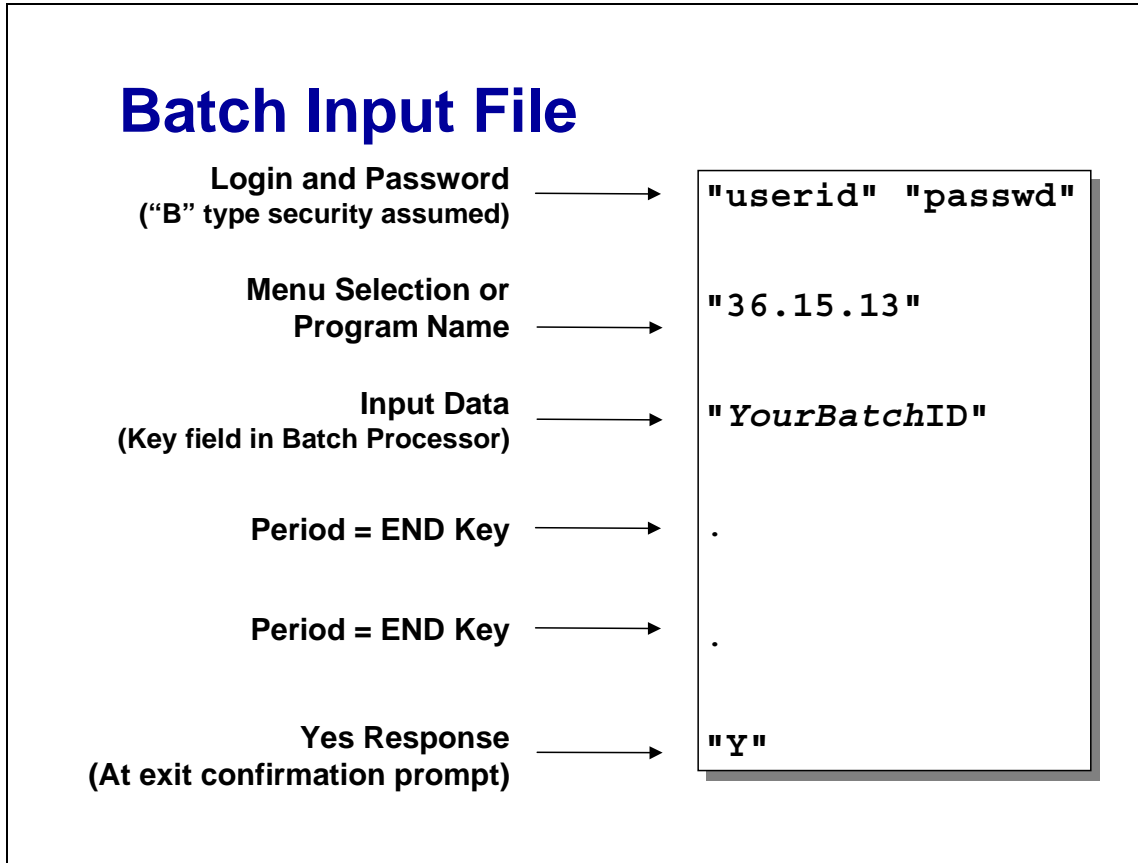
BATRUN.P

```
Propath = "./eBDir,/eBDir/chrcli/us/xrc".  
Input from /WorkDir/bat.in.  
Output to /WorkDir/bat.out.  
Run /eBDir/mf.p.  
Input close.  
Output close.
```

PROGRESS Batch Program



See *Creating a Batch Program* in the *PROGRESS System Administration Reference, Progress Version 9*



Batch Input File



See *Setting Up cron to Run the Batch Program* in the *PROGRESS System Administration Reference, Progress Version 9*

Batch Output File

- Batch mode client session output written to bat.out
 - All standard terminal output is logged
 - File will grow continuously if the process loops
- MFG/PRO function may also cause output
 - File Output
 - Printer Output



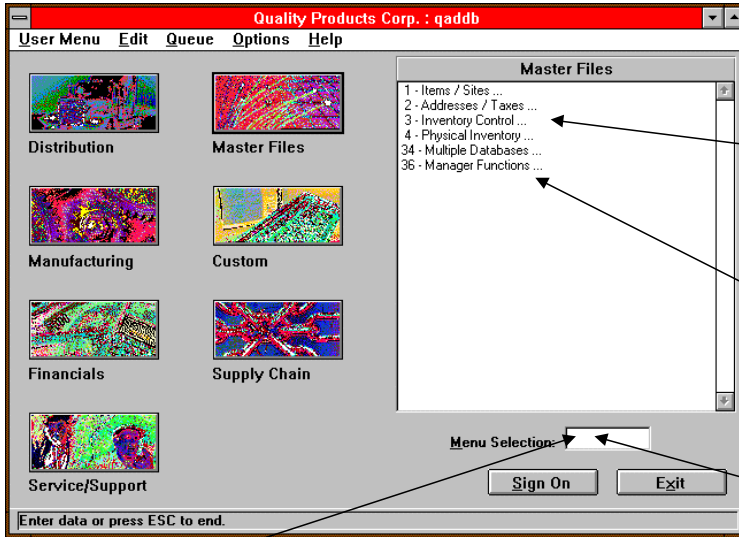
Complete the following Lab: “Scheduling Batch Processes Through cron.”

APPENDIX A

User Interface



Main Menu (Icon Style)



Click on the Master Files button to display these options

Double-click the options to run the programs

or

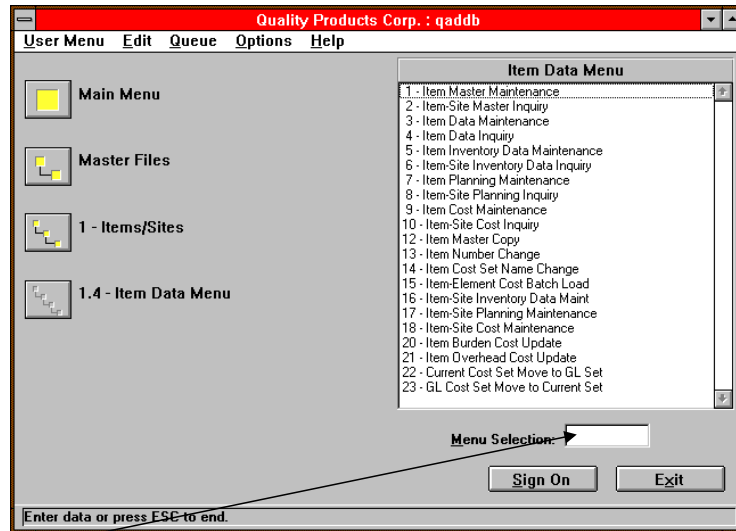
Type the option number here and press Enter

TIP: Use ALT-M to quickly move your cursor into the Menu Selection field

User Interface Overview

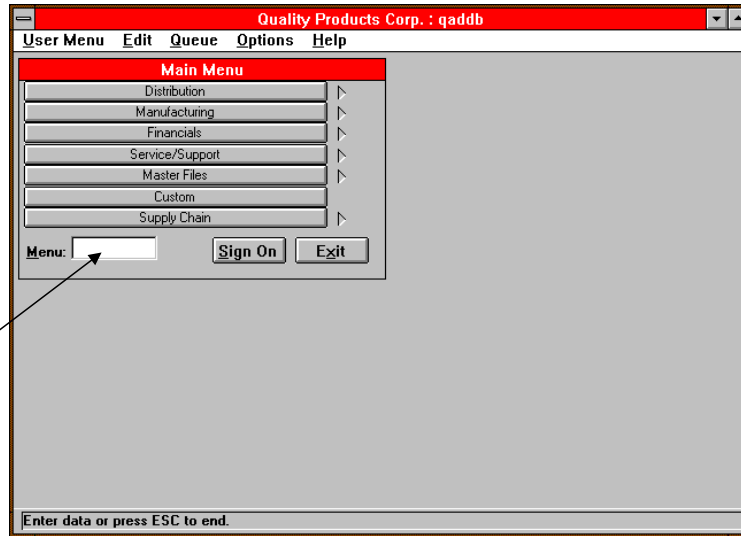
Sub-Menus (Icon Style)

Click these buttons to retrace your steps



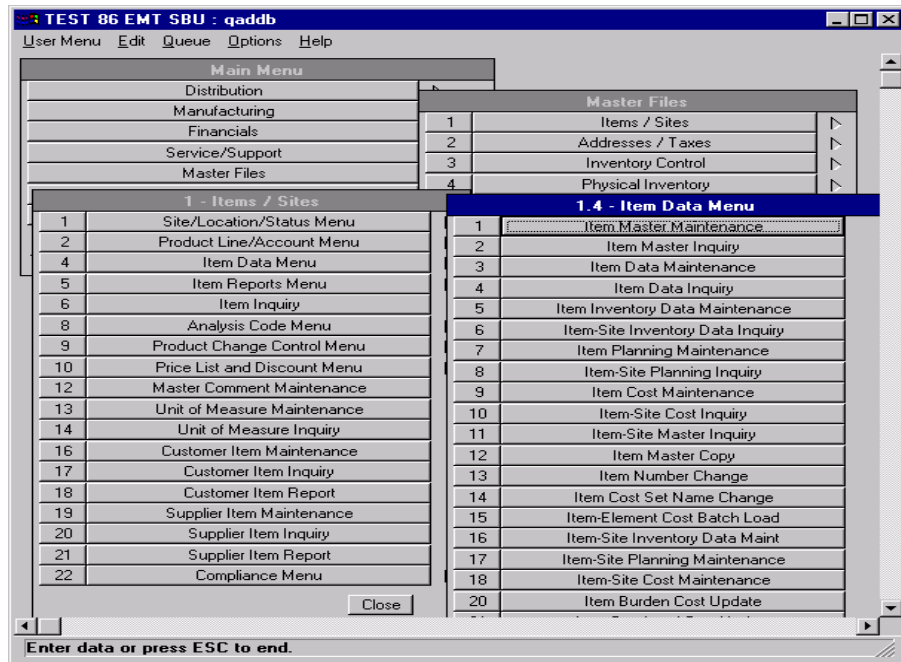
TIP: Using ALT-M lets you jump between modules without first returning to the Main Menu

Main Menu (Tear-Off Style)



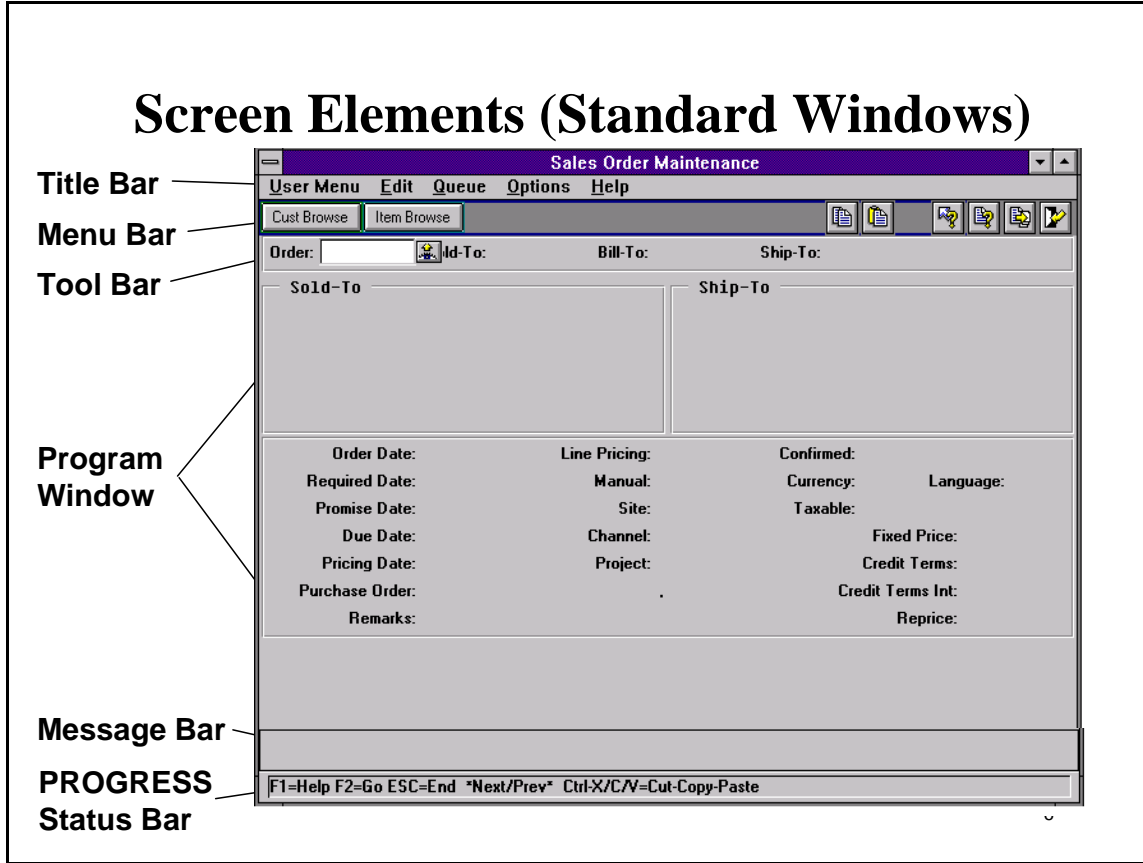
Menu
selection
box

The  symbol means that the option has a submenu

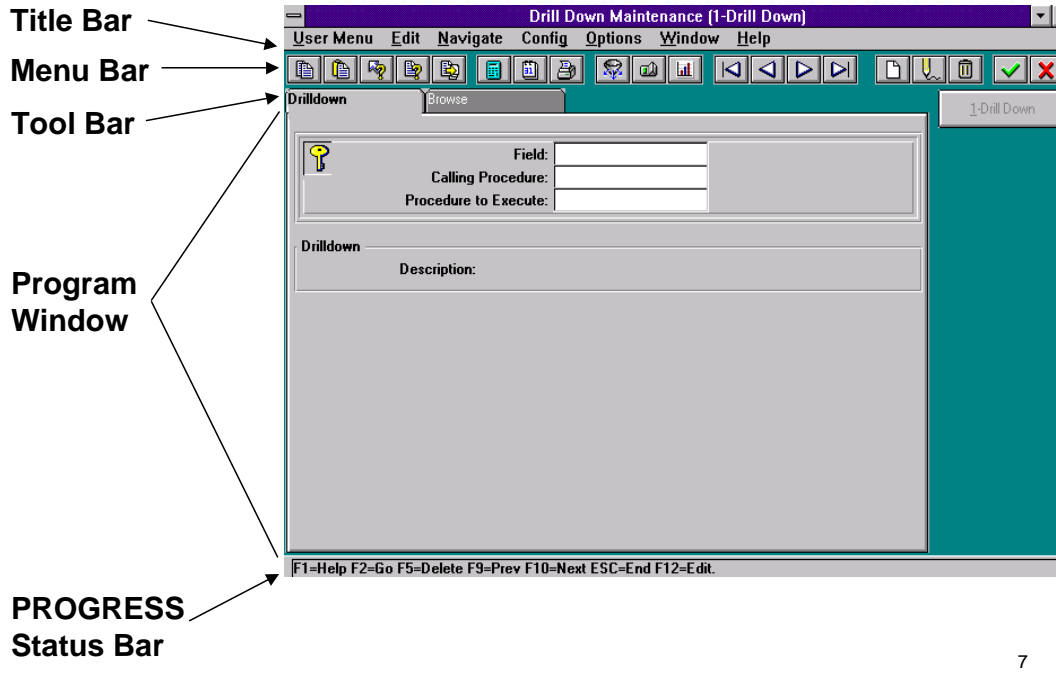


TIP: Only the Tear-Off style expands to use the whole window. This lets you arrange multiple sub-menus on the full desktop area.

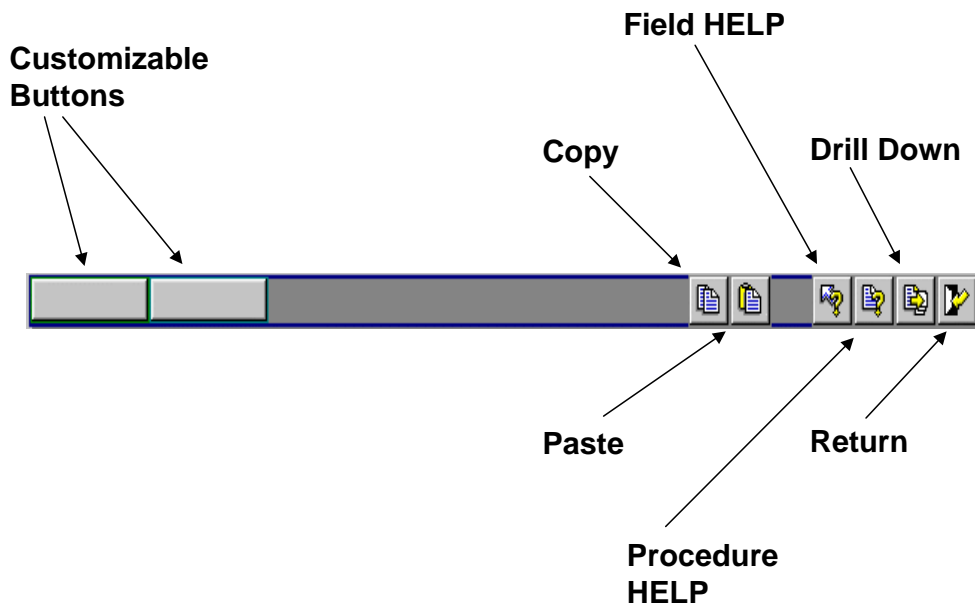
Screen Elements (Standard Windows)

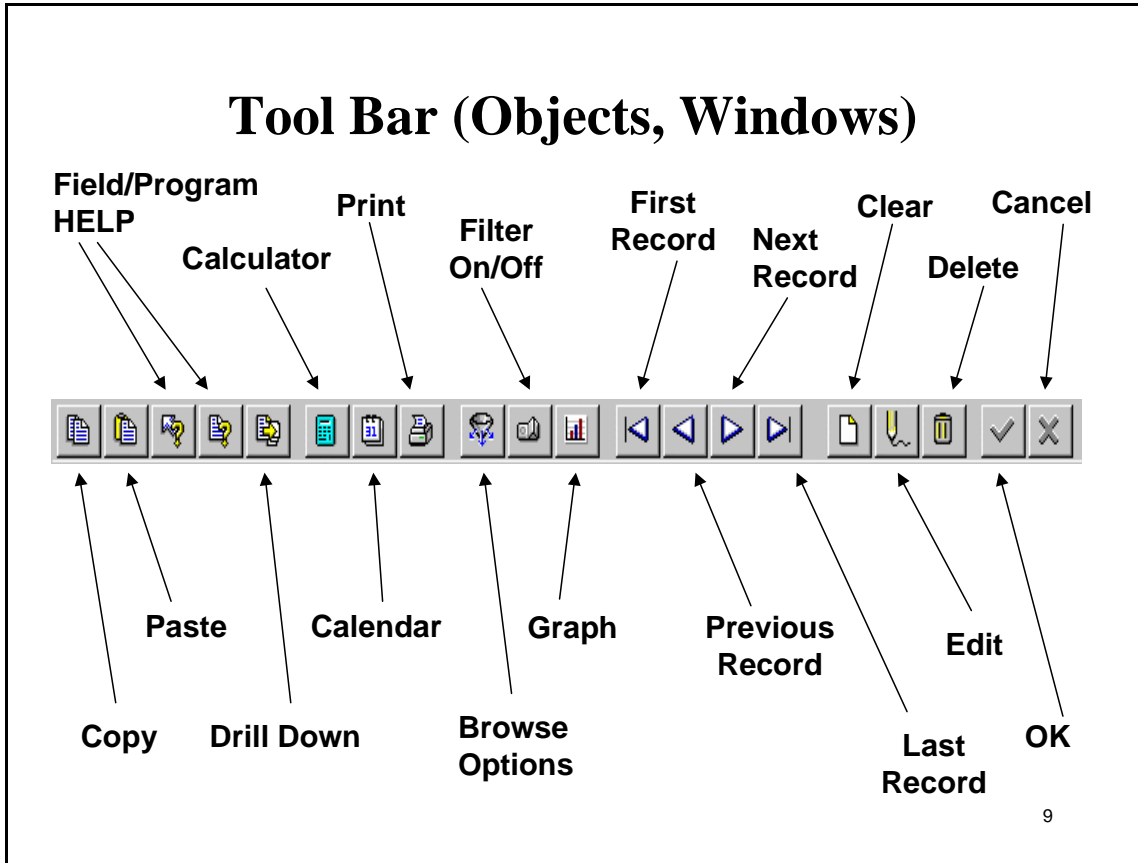


Screen Elements (Objects, Windows)



Tool Bar (Standard Windows)



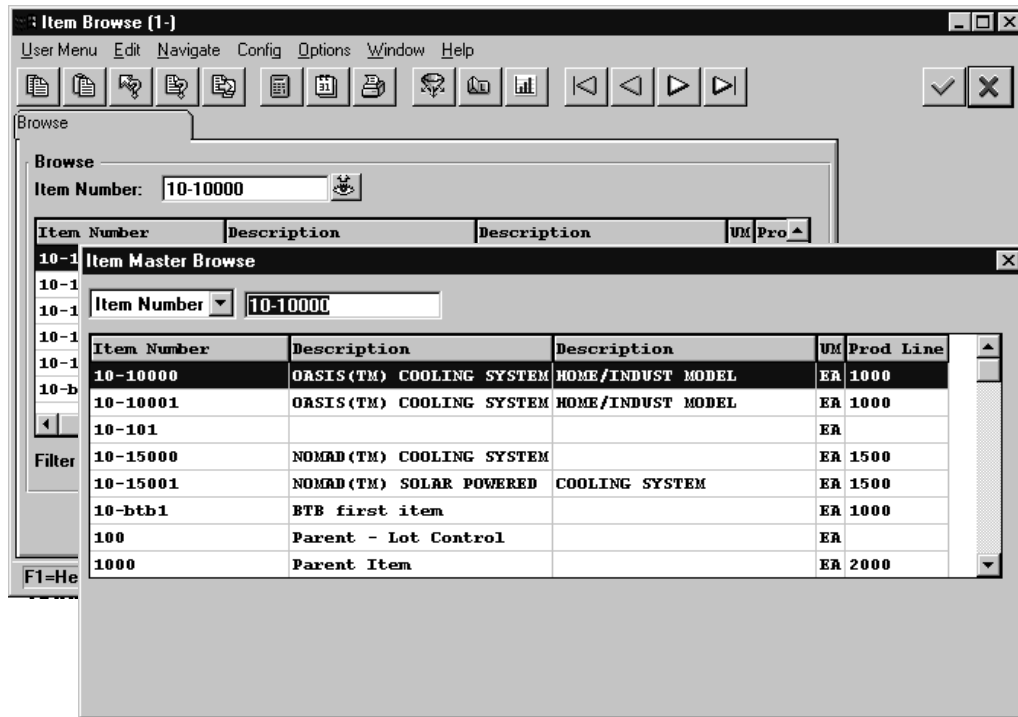


Browsets

What Is A Browse?

Browsets are used as a User Interface to display the results of a query. They return the value you select to the active field in the calling program. By default, the records sort in ascending order on the first column and return the field value of the first column. You can change the default column and order in browse options. The field values in a 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.

1.6 -- Item Browse and Lookup (two kinds of Browsers)



Types Of Browsers

There are two types of browsers: Look-Up Browsers and Power Browsers.

Look-Up Browsers

The Look-up browse is designed to simply display a query of valid data associated with a field. It also returns the value you select to the active field calling the browse.

- It can display and sort up to seven columns of fields
- A Look-up Browse cannot run independently; i.e., the Look-up must be linked to a field (see *Linking a Look-Up Browse* on page 168)

Accessing a Look-up browse

There are three ways you can access a Look-up in GUI. Tab to the field you want to browse and then do one of the following.

Click on its icon next to the field that it is attached to.

or

Click on the Help menu and drag down to Look-up Browse.

or

Press Alt + F2.

To access a Look-up in character tab the field you want to browse and then press F2.

Power Browsers

Power Browsers have the capability of doing much more than Look-ups. They feature the full functionality for Sorting, Filtering (range criteria on first 10 fields), graphing, printing, and exporting data.

- They can display up to 20 columns of fields (a maximum of 330 characters)
- The first seven columns can be sorted, the first 10 columns filtered
- Power browses can be linked to a field or run independently

For more information on the features of a Power browse, refer to pages 58-65 of the MFG/PRO 8.5/8.6 User Interface Guide.

Accessing a Power browse

There are two ways you can access a power in GUI. Tab to the field you want to browse and then do one of the following.

Click on the Help menu and drag down to Drill Down.

or

Press Alt + F1.

Other ways of running a Power Browse

Unlike Look-Up browses, Power browses can run as an independent application. With power browses you can:

- Assign it a menu number to replace an existing menu item or add a new menu item

- Put a power browse on the tool bar as a button
- Embed it in an OBCM maintenance application as a Browse folder
- Attach it to a field as a drill-down

Requirements are:

- Fields to display: Item Number, Description, and Unit of Measure
- Table/View we want to query: Item Master (pt_mstr)
- Returned value: Item Number (pt_part)
- Sort by: Description (pt_desc1)

pt_mstr

Field Name	Label	Format
pt_part	Item Number	x(18)
pt_desc1	Description	x(24)
pt_um	UM	x(2)

Designing A Browse

Before creating a browse you must first determine the requirements. What type of data do you need to query? Why? How is it used? What additional fields need to be displayed? What field value is returned? How should the data be sorted? All these questions must be answered before a browse can be created.

Let's look at an example using the above requirements. When creating Purchase Orders in MFG/PRO, you need to enter valid line item parts (that is, parts that are in the Item Master database). In this case, you may want to have a browse that tells you all of the purchased parts that are in the system. Chances are you also want to know the description of the part as well as the Unit of Measure. Let us suppose Procurement has a unique requirement in that they would like to see the available items sorted by the description of the part.

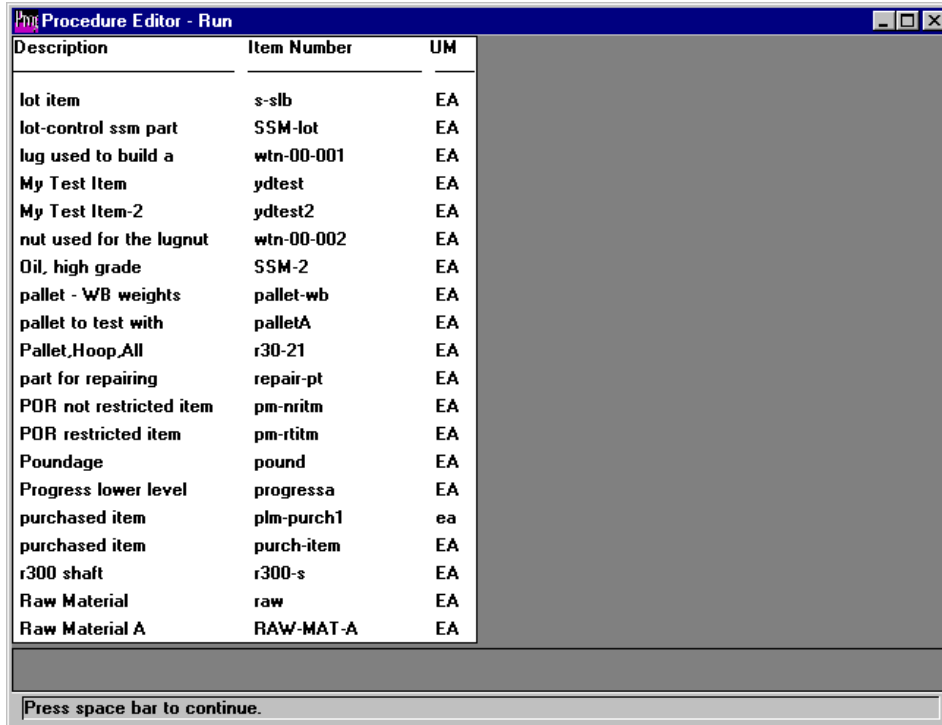
Before we go in to the mechanics of how to create a browse, it is important to understand that a browse is basically a FOR EACH loop with a WHERE clause. The following Progress code generates the desired output.

```
for each pt_mstr where pt_pm_code = "P" by pt_desc1:
```

```
    display pt_desc1 pt_part pt_um.
```

```
End.
```

Procedure Editor - Run



Description	Item Number	UM
lot item	s-slb	EA
lot-control ssm part	SSM-lot	EA
lug used to build a	wtn-00-001	EA
My Test Item	ydtest	EA
My Test Item-2	ydtest2	EA
nut used for the lugnut	wtn-00-002	EA
Oil, high grade	SSM-2	EA
pallet - WB weights	pallet-wb	EA
pallet to test with	palletA	EA
Pallet,Hoop,All	r30-21	EA
part for repairing	repair-pt	EA
POR not restricted item	pm-nritm	EA
POR restricted item	pm-ritm	EA
Poundage	pound	EA
Progress lower level	progressa	EA
purchased item	plm-purch1	ea
purchased item	purch-item	EA
r300 shaft	r300-s	EA
Raw Material	raw	EA
Raw Material A	RAW-MAT-A	EA

Press space bar to continue.



Exercise 1

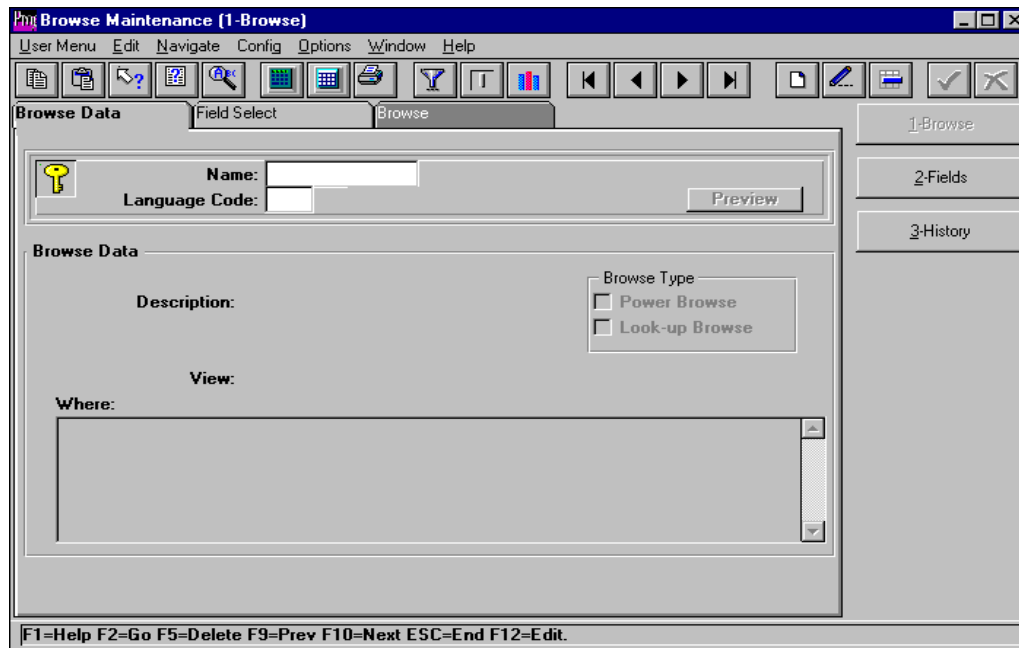
From your progress editor type in the following PROGRESS code and then press F2 to run it. You should see a query of your data similar to the screen above. Note the results of your query because we compare your output with the output from a browse that you create later today.

for each pt_mstr where pt_pm_code = "P" by pt_desc1:

display pt_desc1 pt_part pt_um.

End.

36.18.10 -- Browse Maintenance (1-Browse): Browse Data



How To Create A Browse

Browse Maintenance (mgcn011.p) is used to create, modify, and delete browses.

Use Browse Maintenance to create both a Power browse and a Look-up for the example requirements discussed above.

To start Browse Maintenance enter mgcn011 in the menu selection, or select menu option 36.18.10 in 8.5 or 36.20.13.10 in 8.6. The above screen appears.

- 1 Enter the name of the new browse you are creating.
Use xx123 for your name.
- 2 Enter the Language Code.

US.

3 Choose Go.

4 Enter the Description of your new browse.
Purchased Items.

5 Under Browse Type, select both Power Browse and Look-up Browse.

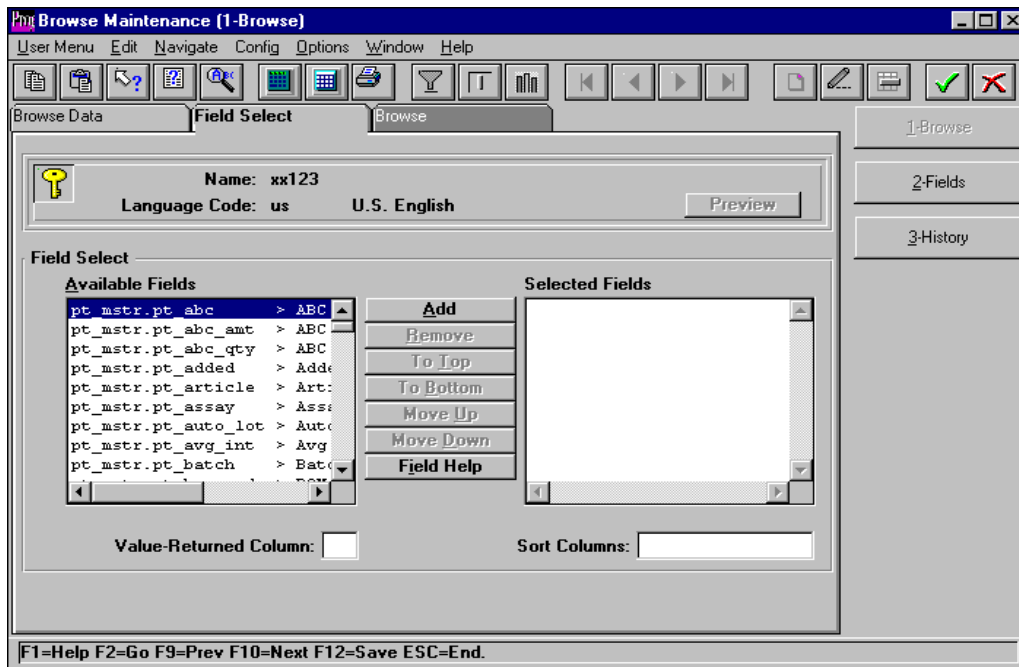
6 Enter the View.
pt_mstr

36.18.10 -- Browse Maintenance (1-Browse): Browse Data



- 7 Enter the Where clause.
pt_pm_code = "p"

36.18.10 -- Browse Maintenance (1-Browse): Field Select



- 8 Choose Go to continue. The above screen appears.
- 9 Select the fields to be displayed by highlighting each desired field and pressing the Add button. Select pt_desc1, pt_part, and pt_um respectively.
- 10 Enter the Value_Returned Column. The default returned value is the first column. Since we want the part number to be the returned value, we must select 2.
- 11 Enter the Sort Column. The default sort column is the first column. Since we want this browse sorted by the description, we should leave this value blank.
- 12 Choose Go to continue.

36.18.10 -- Browse Maintenance (2-Fields): Browse Field Data

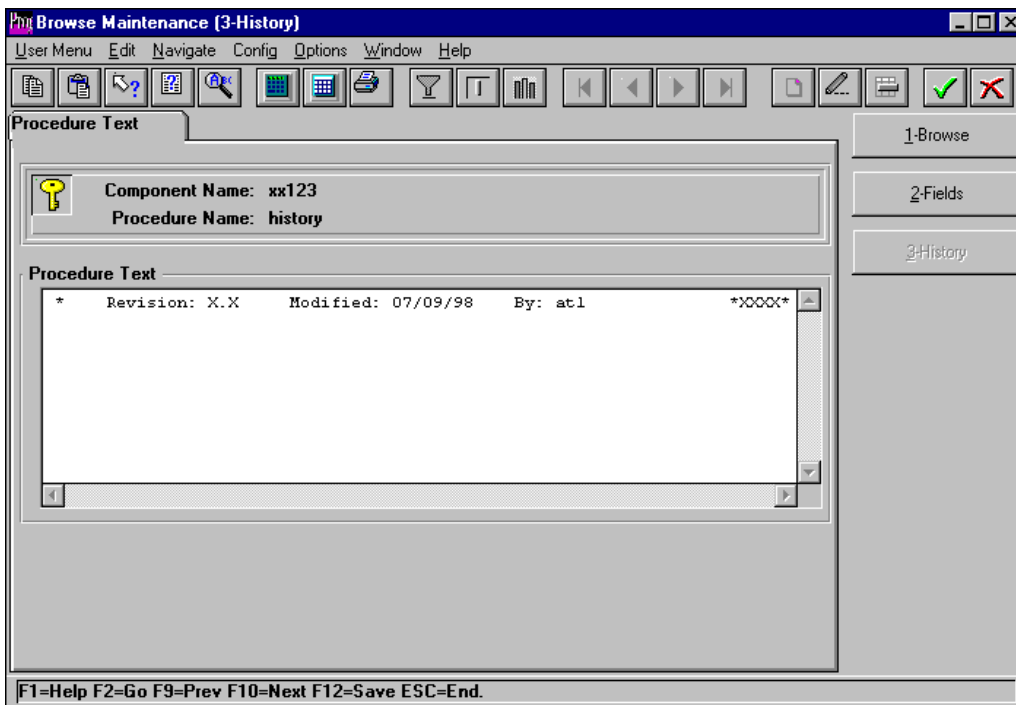


- 13 At this point we have completed everything that needs to be done in the 1-BROWSE application. The next step in creating a browse is to click on the 2-FIELDS button. The above screen appears once you have selected your Language Code and press Go to proceed.

This screen is used to override the default label and the format if you desire. For example, if you wanted to change the label for pt_desc1 to desc and truncate the format to 20 bytes, you enter those values in the respective fields. When you are done with your updates, press Go to proceed.

- 14 The next step is to click on the 3-HISTORY button.

36.18.10 -- Browse Maintenance (3-History): Procedure Text



15 Press Go, and you see the above screen.

This screen is used to record the modification log of the browse you created. An entry should be made when the browse is created. Also, an entry should be made every time the browse is modified. This historical data gets transcribed on the Progress code created from Browse Maintenance.

- In the Procedure Text replace "X.X" with the version of MFG/PRO you are running.
- When you are through editing your Procedure Text, press the green check-mark button.
- Now press the 1-BROWSE button to get back to the main screen.

Note In order for your history log changes to take effect you must press Go three times from the 1-BROWSE application. Code gets generated only from the 1-BROWSE container.

You are now finished creating your browse! You may now preview your browse by pressing the Preview button from the 1-BROWSE application. If you created a Power Browse and a Look-Up Browse, both browses are launched (first the Power Browse, then the Look-up appears after the Power Browse has been closed).

Browse Naming Conventions

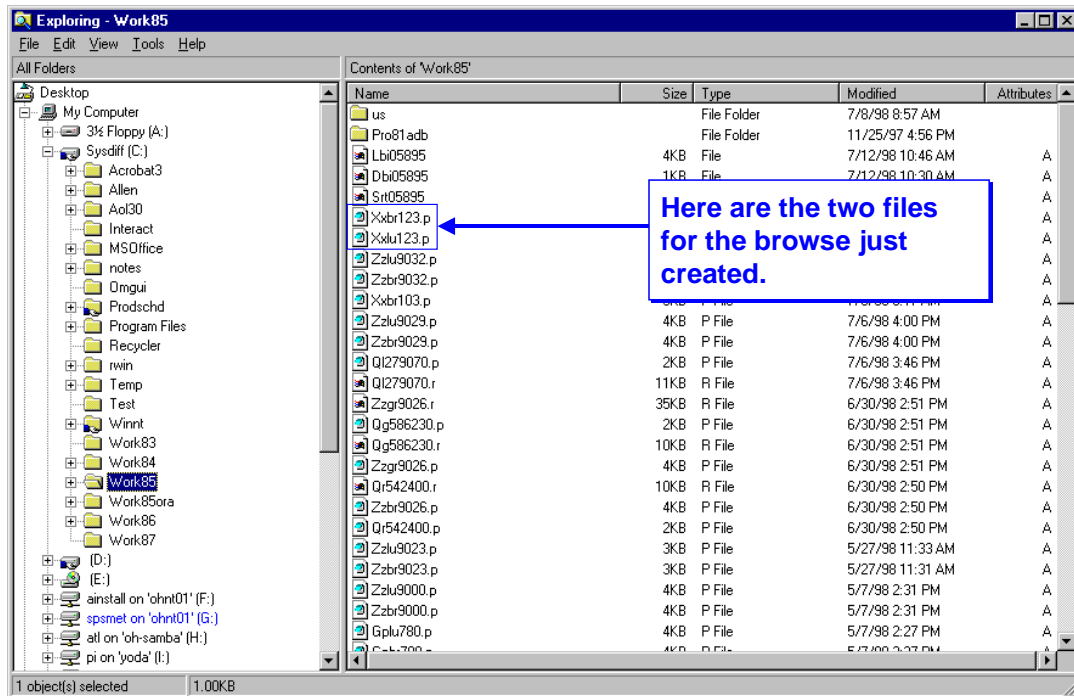
<u>Browse Name</u>	<u>Progress Source Code Created</u>	<u>Type of File</u>
xx123	xxbr123.p	Power Browse
	xxlu123.p	Look-Up Browse

- The first two characters represent the module (ar – Accounts Receivable, so – Sales Order, gp – General Purpose, etc.). We encourage our customers to use **xx** for custom browses. This will eliminate conflicts with browses created by QAD.
- The next two characters describes the type of browse. **br** – Power Browse, **lu** – Look-up Browse.
- The last three digits should be uniquely sequenced to identify the browse.

A Note about Browse Names

You have just created two new Progress programs as a result of the previous steps. The names of the files are based on the name of the browse given. The browse generator creates the above source code.

Exploring - Work85



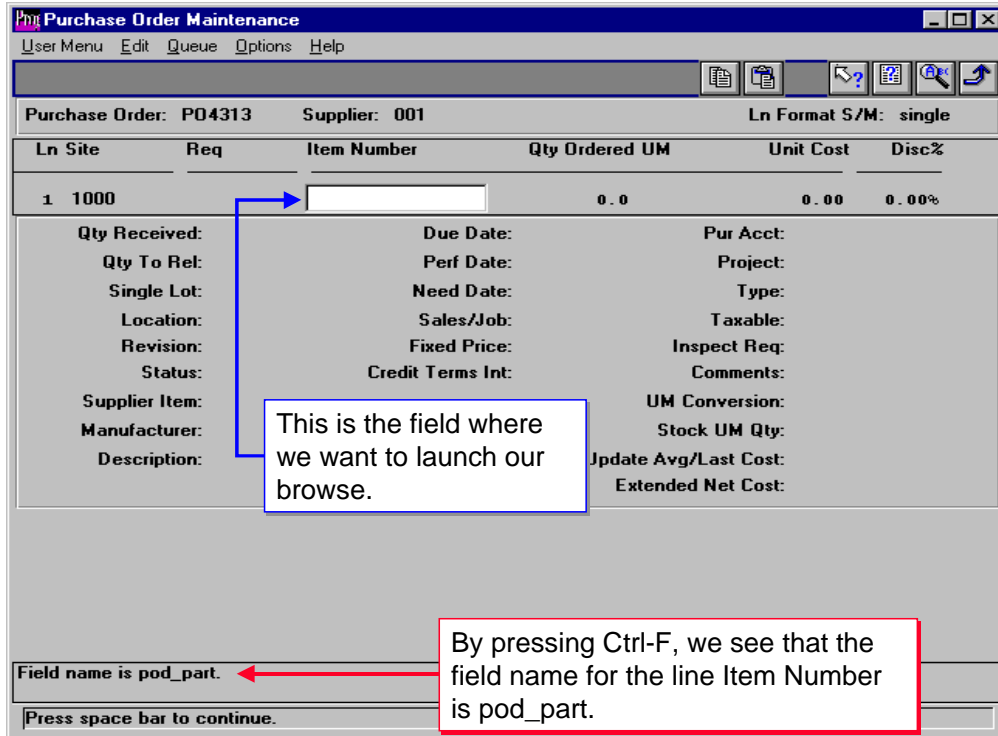
Now that we have created our browse, let's find the source code using Windows Explorer. They should be in the work directory that MFG/PRO is running on.



Exercise 2

- 1 Given the requirements described on page 5, use Browse Maintenance to create a Power browse and a Look-up browse.
- 2 Run your browse using the Preview button. How does your output compare with your output from Exercise 1?
- 3 Using Windows Explorer, locate the Progress source code created by the browse generator. The file names you are looking for are `xnbr123.p` and `xxlu123.p`. These files should be located in the directory you are running MFG/PRO from.

5.7 -- Purchase Order Maintenance



Linking Your Browse

In most cases (all cases for Look-Ups), the intended use of browses is to have them launched from a specified field on a screen. Let's revisit the example browse we created earlier. What was the original purpose for creating this browse? We wanted to see all the valid Purchased Parts in the system. So at what point do we want to launch this browse? Answer: when we are in the Purchase Order Maintenance screen (popomt.p - 5.7) at the point we entering line item numbers. Specifically when we are on the line item number field (pod_part).

There are three things we need to know before we can create our link.

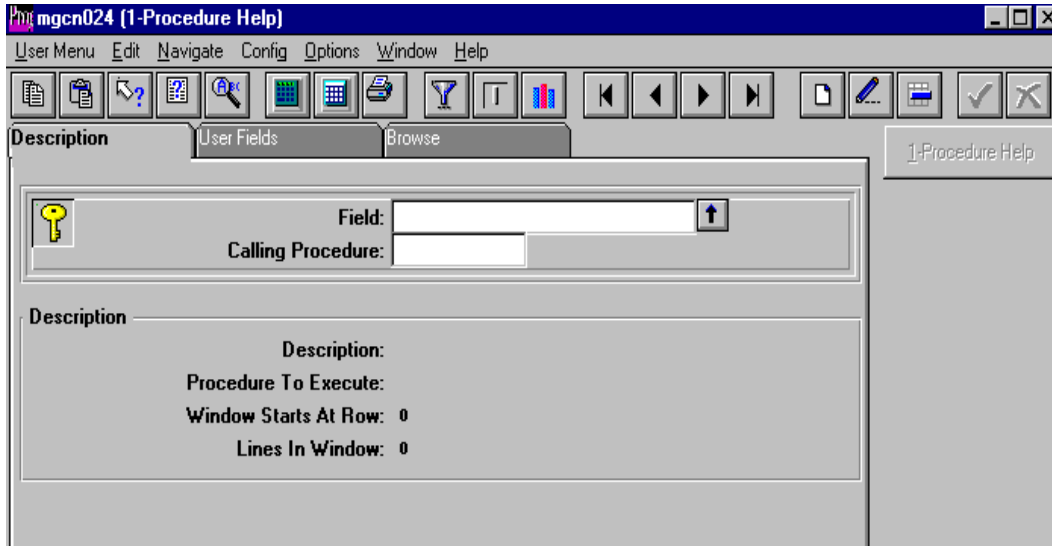
- What is the **Field** we are launching the browse from?

- What is the **Calling Procedure**? Note that we do not have to specify a calling procedure. If a calling is not specified, then this browse is launched every where in the system the specified field is found.
- The **Procedure to Execute** specifies the browse we want to launch. Pressing Ctrl-F1 gives you the name of the procedure.

So in the case above we have:

Field:	pod_part
Calling Procedure:	popomt.p
Procedure to Execute:	xxlu123.p for Look-up browses and xxbr123.p for power browses

36.7.21 -- mgcn024 (1-Procedure Help)



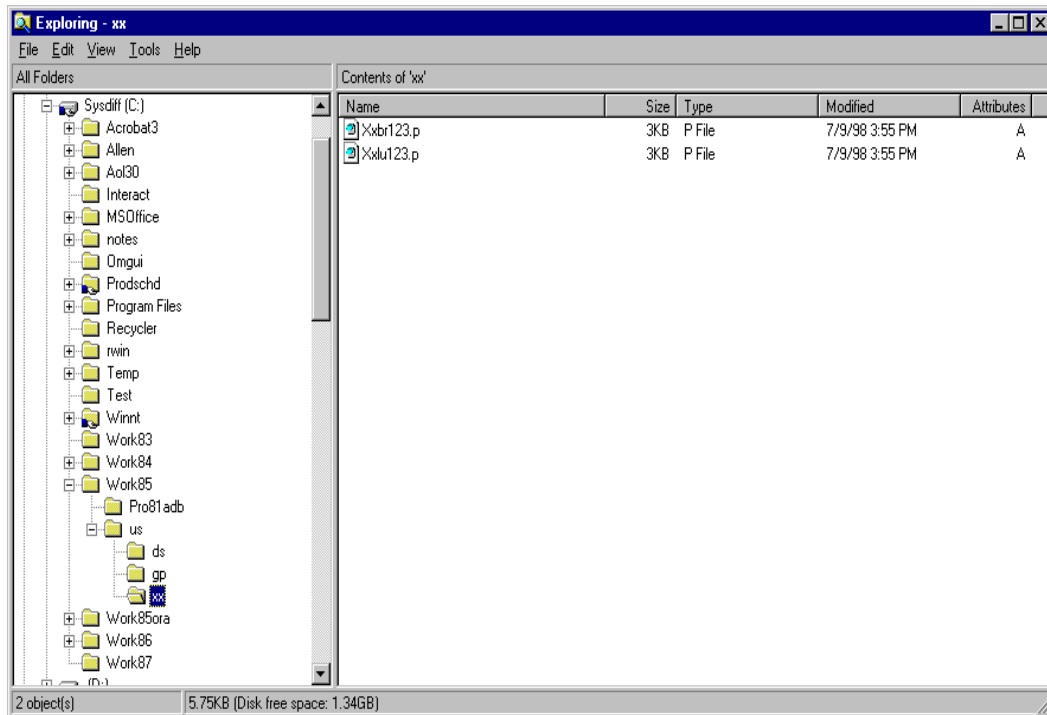
Linking a Look-Up Browse

Window Help Maintenance (mgcn024.p - 36.7.21) is used to link Look-Up browses to a field. Field links can be global or specific to a particular screen.

Let's create a link for our case described in the previous page.

- 1 Run Window Help Maintenance and you see the above screen.
- 2 Enter the Field and the Calling Procedure. Then press Go.
- 3 You then need to enter the Description and the Procedure to Execute.
- 4 Press Go, then exit after you have entered your values.

Exploring - xx



Your link is now complete. However, before you can test it, you must make sure that your source code is in the proper directory. By default, the source code you create is saved in the folder of your local drive where MFG/PRO runs. In our example, the browse we created was stored in c:\work85. In order for our link to work, you must create the subdirectories c:\work85\us\xx where xx is the first two characters of your browse name. Then you need to move your browse source code into the xx folder.

Linking a Power Browse

Drill Down Maintenance (mgcn001.p - 36.20.1) is used to link Power browses to a field. The steps you need to take are similar to what is described in the previous page.



Exercise 3

- 1 Use Window Help Maintenance (mgcn024.p) to link your Look-Up browse to the Purchase Order line Item Number (see page 3 for specifications).
- 2 Use Drill Down Maintenance (mgcn001.p) to link your Power browse to the Purchase Order line Item Number.
- 3 Use Windows Explorer to create the us\xx folders and then move your browse source code into the xx folder.
- 4 Test both, your Power browse and your Look-up browse link by going into Purchase Order Maintenance launching both browses from the line Item Number field.

Views

What Is A View?

A view is a display of all of the fields and locally defined variables 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.

Designing a View

ld_det - Inventory Detail (by Location)

Field Name	Label	Format
ld_part	Item Number	x(18)
ld_loc	Location	x(8)
ld_qty_oh	Qty On Hand	->,>>>, >>9.9<<<<<<<<
ld_qty_all	Qty Allocated	->,>>>, >>9.9<<<<<<<<

pt_mstr - Item Master

Field Name	Label	Format
pt_part	Item Number	x(18)
pt_desc1	Description	x(24)

Variable

```
local-var01 = ld_qty_oh - ld_qty_all
```

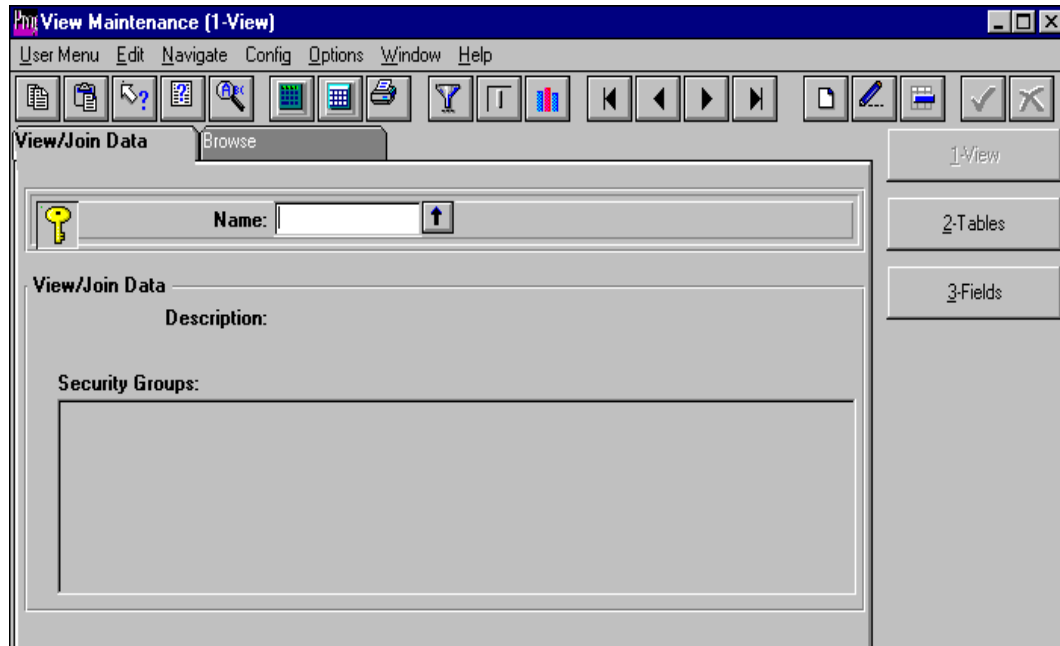
Designing A View

Suppose you wanted to create a browse that displays the following inventory information.

- Item Number
- Location
- On Hand Quantity

This can be accomplished by selecting the Inventory Detail (ld_det) table when creating your browse. But what if you wanted to display a calculated field that is not stored on the table, like Qty Unallocated (Qty On Hand - Qty Allocated). And what if you also wanted information that is not on the Inventory Detail table, like Item Description? This can be accomplished by creating a view that joins the above two tables and defines a local variable for Qty Unallocated.

36.20.12.12 -- View Maintenance (1-View)

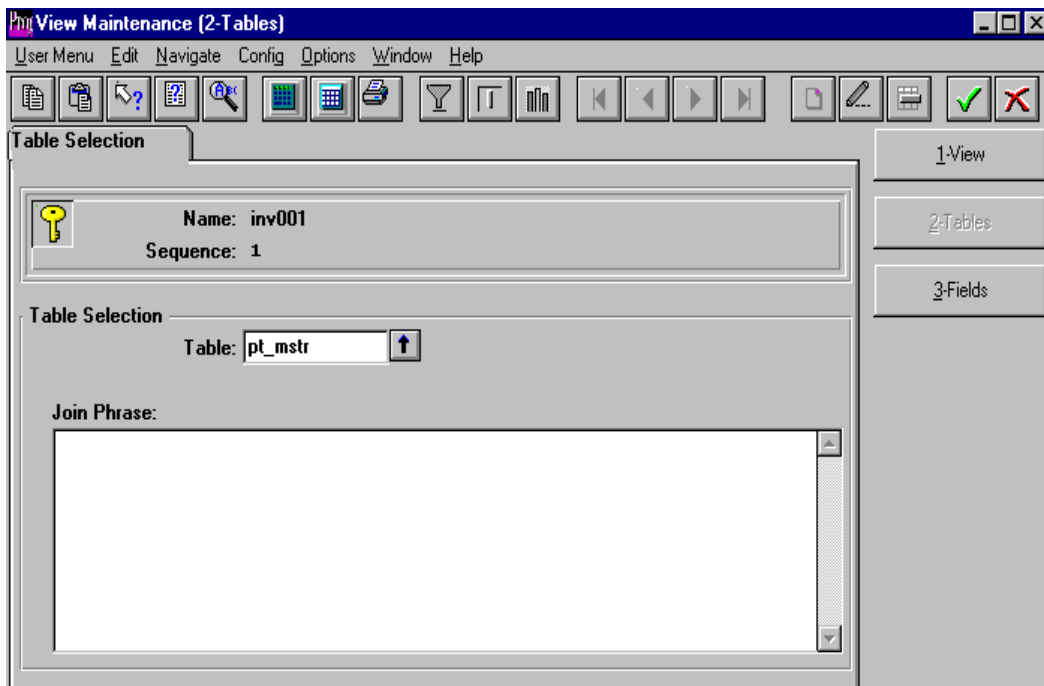


How To Create A View

Use View Maintenance (mgcn008.p) to create or modify views. Let's step through creating a view for the scenario described above.

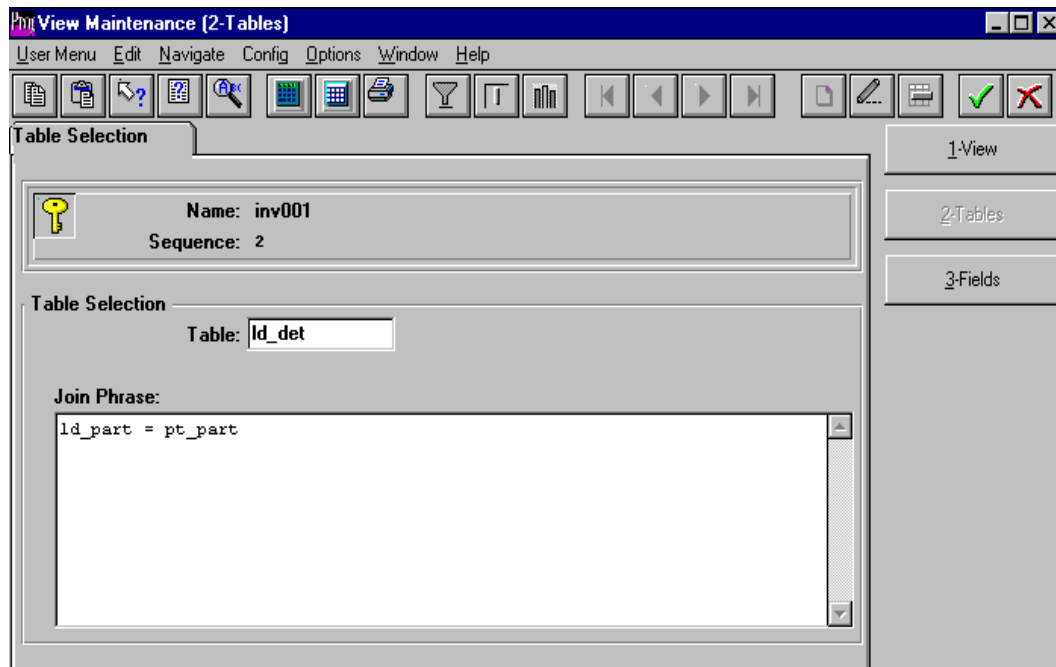
- 1 Select View Maintenance (36.18.12 in 8.5 and 36.20.12.12 in 8.6). The above screen appears.
- 2 Enter the view name.
inv001
- 3 Press Go. Then enter the description of the View.
Inventory Unallocated

36.20.12.12 -- View Maintenance (2-Tables)



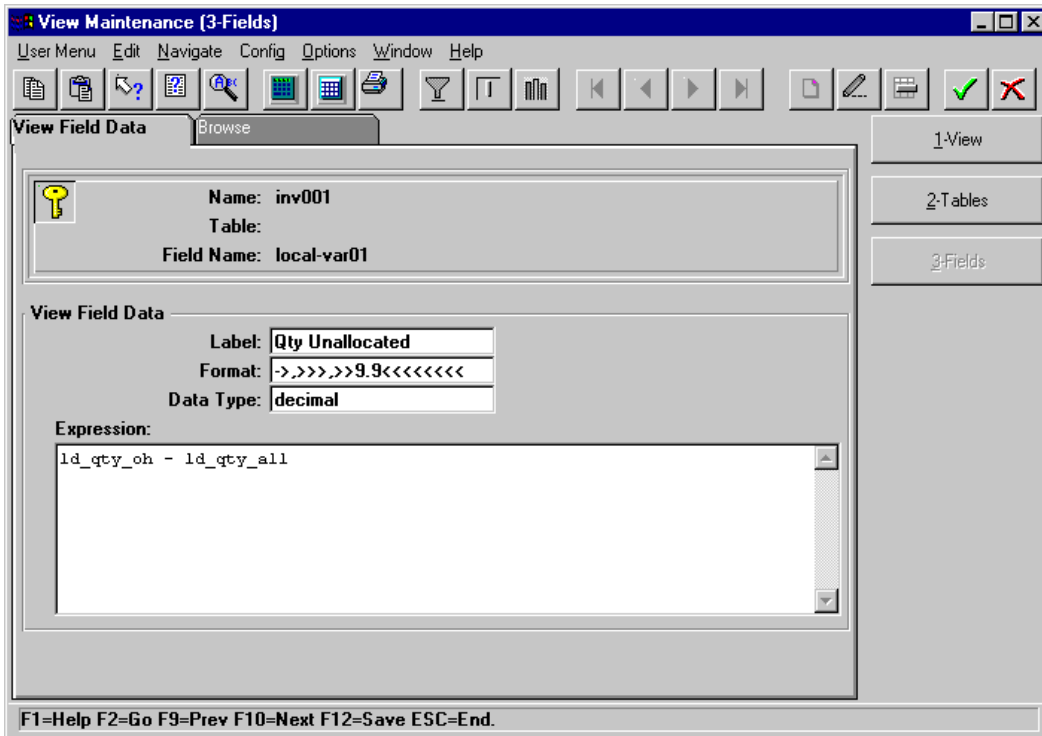
- 4 Select the 2-TABLES button and then press Go.
- 5 Enter one of the tables needed then press Go. Enter pt_mstr.
- 6 Now select 2 for Sequence and press Go.

36.20.12.12 -- View Maintenance (2-Tables)



- 7 Enter the other required table (ld_det) and then tab down to the Join Phrase. You need to join the two tables with a common field. Entering ld_part = pt_part does this.
- 8 Press Go.
- 9 Select the 3-FIELDS button.

36.20.12.12 -- View Maintenance (3-Fields)



10 Here is where our local variable is defined for unallocated quantity. To do this, leave the table name blank, enter the local variable and then press Go. When entering a local variable, name it local-varnn, where nn is a number incremented by one from the last defined variable.

11 Now enter the values as shown on the next page to define your local variable.

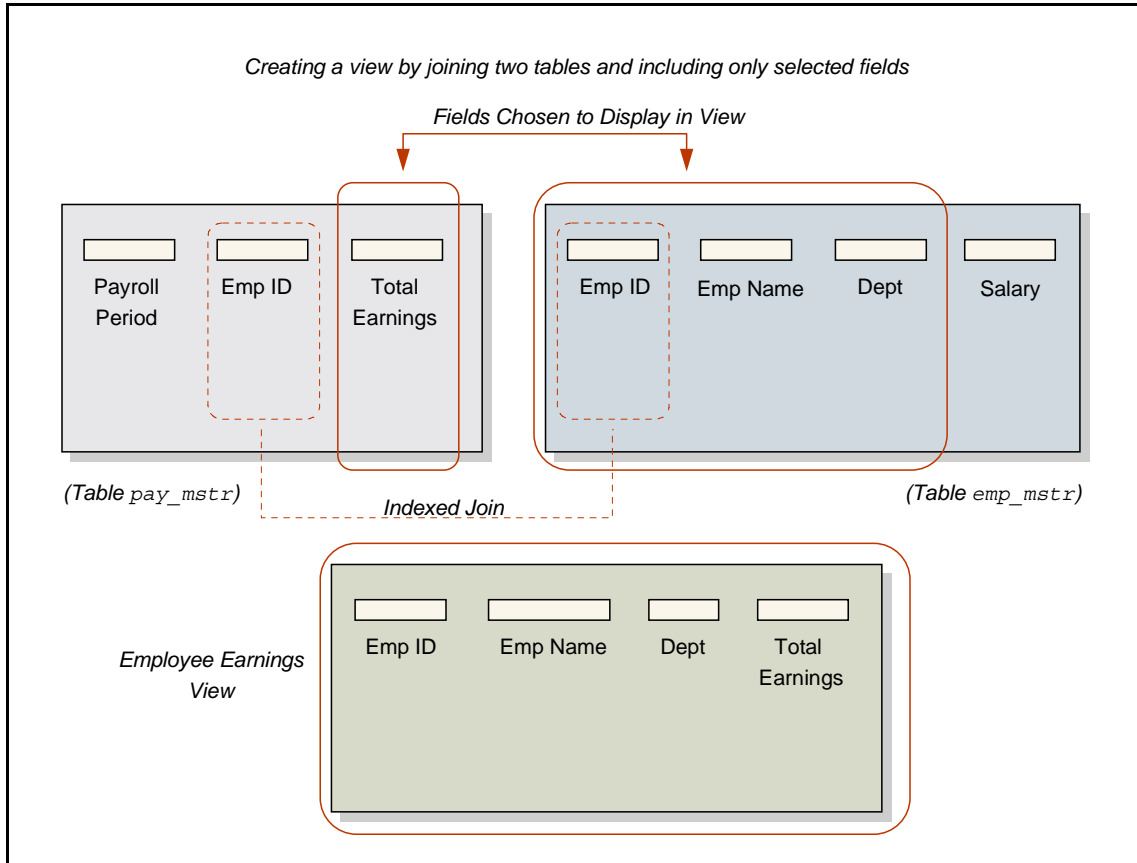
12 Press the green check mark to save your changes.

Your view is complete and can now be used when creating a browse.



Exercise 4

- 1 Use View Maintenance (mgcn008) to create the view described above.
- 2 Create a Power browse using Browse Maintenance that displays the following fields from view inv001. Refer to page 172 for the exact name of the fields.
 - Item Number
 - Description
 - Location
 - Qty On Hand
 - Qty Allocated
 - Qty Unallocated



Creating a view on a single table to restrict fields available to be used in other views or browses

Employee Department List

Emp ID	Emp Name	Dept	Salary

(Table emp_mstr)

Character Environment: mgcn011.p (1-Browse)

```

User Menu Edit Navigate Config Options Window Help
-----
mgcn011.p (1-Browse)                                07/16/98
-----
Name: xx125                                           <1-Brows>
Language Code: us   U.S. English                       <2-Field>
                                           <3-Histo>
                                           <Preview >
-----
Browse Data
-----
Description: 
View: 
Where:



Browse Type
[X]Power Browse
[ ]Look-up Browse
-----
F1=Go 2=Help 4=End ESC-m=Menu 6=UserMenu 9=Prev 10=Next 12=Save

```

Character Environment

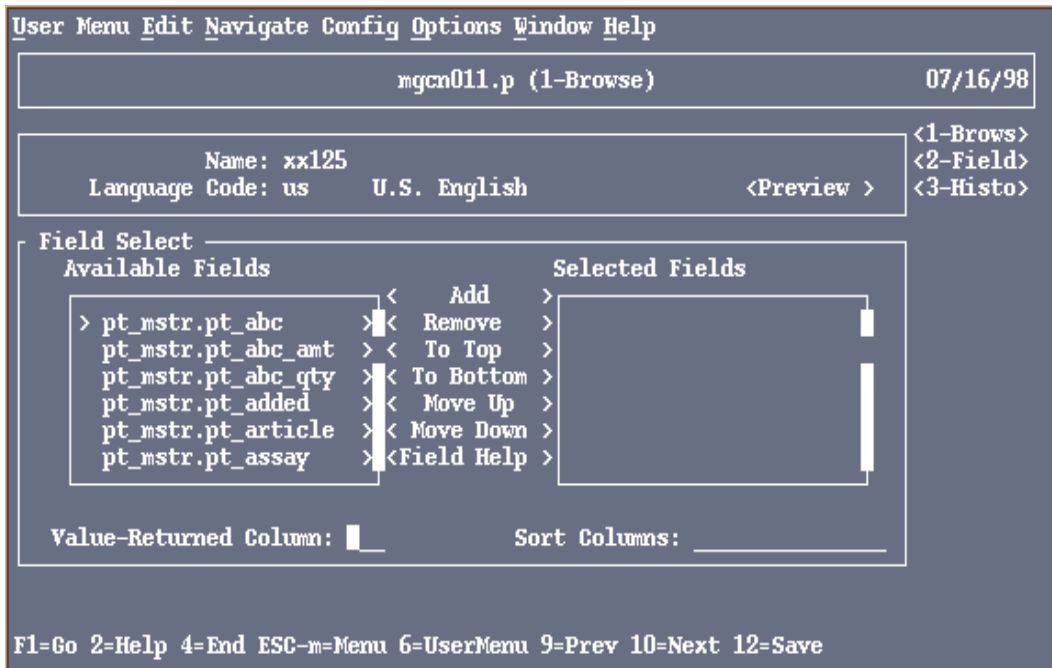
Working with browses in the character environment is not very user friendly. It is especially challenging to navigate around Browse Maintenance. For this reason, it is recommended that browse maintenance be done in the Windows environment. Unfortunately there may be no way to get around this because there are a number of customers running MFG/PRO in the character environment. For this reason we walk through creating a browse in the character environment.

Recreate the browse from Exercise 2, but call it xx125 (since xx123 and xx124 are taken).

- 1 Start your character session and go into Browse Maintenance (mgcn011.p).
- 2 Enter the browse name (xx125) and the language (us).
- 3 Press Go (F1). The above screen appears.

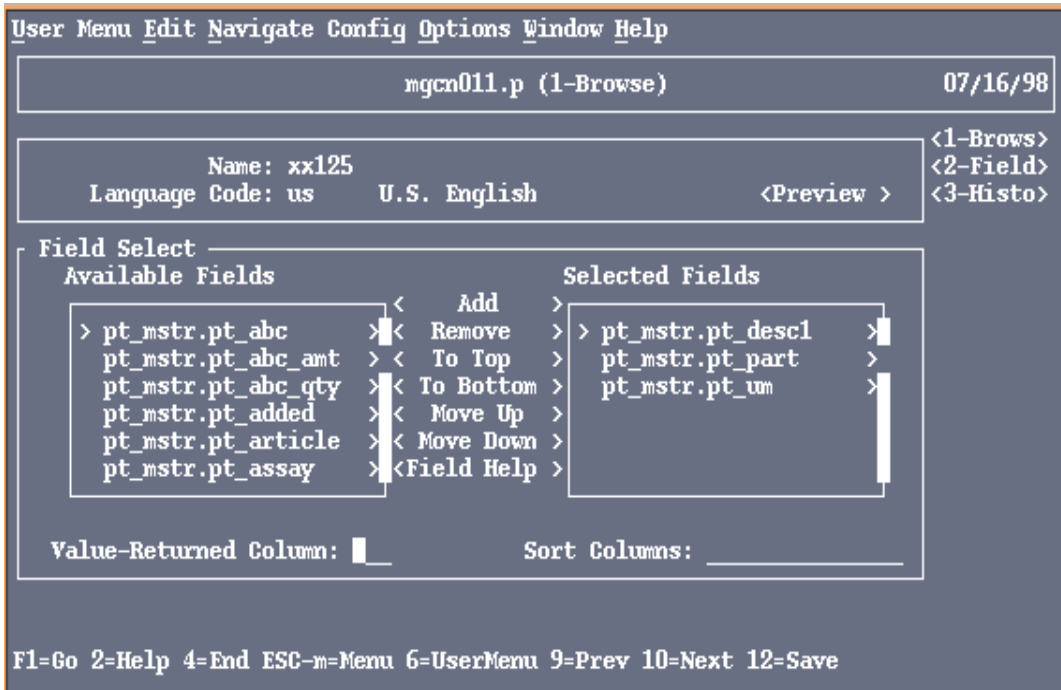
- 4 Enter the Description and tab over to the Browse Type box. You notice that Power Browse by default is selected. Press tab one more time to the Look-up Browse selection. Press the space key to select Look-up Browse also.
- 5 Tab to the View field and enter pt_mstr
- 6 Tab to the Where field and enter pt_pm_code = "p"

Character Environment: mgcn011.p (1-Browse)



7 Press Go and the above screen appears.

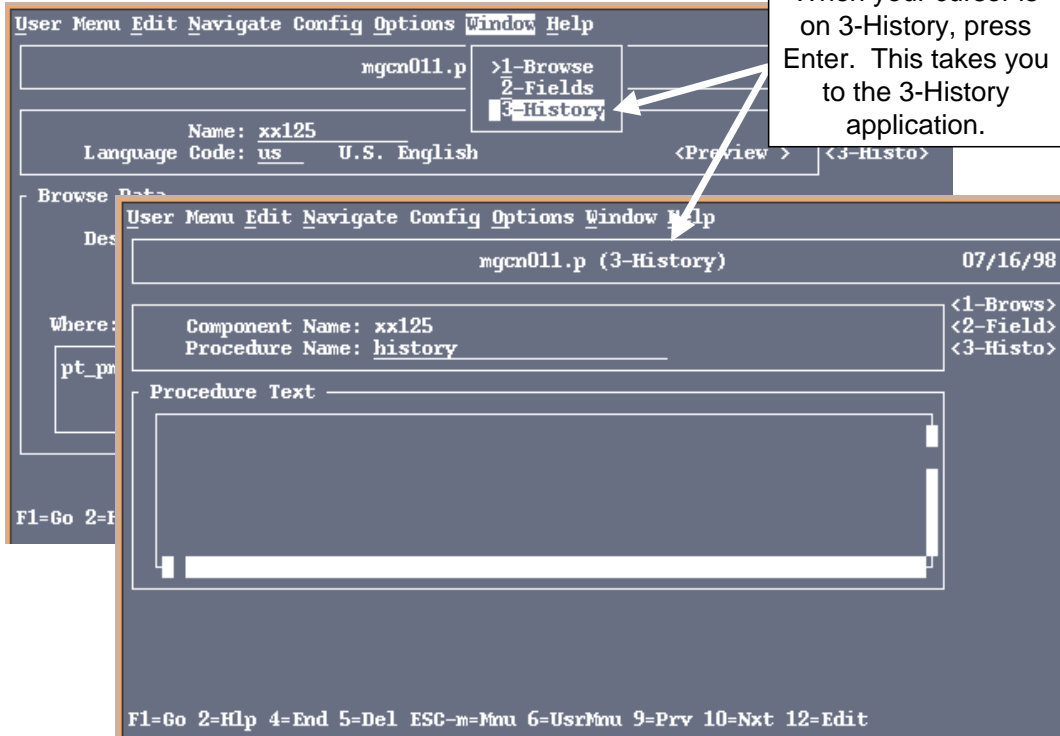
Character Environment: mgcn011.p (1-Browse)



- 8 To select your fields, tab to the **Available Fields** box. Press space to mark the field you wish to select and tab to the **Add** operation. Press Enter while your cursor is on Add and your selected field is moved to the **Selected Fields** box. Repeat until all of the desired fields have been selected.

You need to press Go twice to get back the Value-Returned Column. Enter 2 because we want the Part number to be returned from the browse. Then Press Go to save your changes.

Character Environment: mgcn011.p (3-History)



- 9 To navigate to the 2-FIELDS application or the 3-HISTORY application, press ESC-m to invoke the menu and press the right arrow key as many times as it takes to get to the Window menu. Press down arrow to the 3-HISTORY application so we can create a history log of this browse.
- 10 Press Go to get into the Procedure Text and enter your history log.
- 11 Press Go to save your changes.
- 12 To regenerate your code you need to go back to the 1-BROWSE application and press Go twice.

You are done creating your browse in the character environment!



Exercise 5

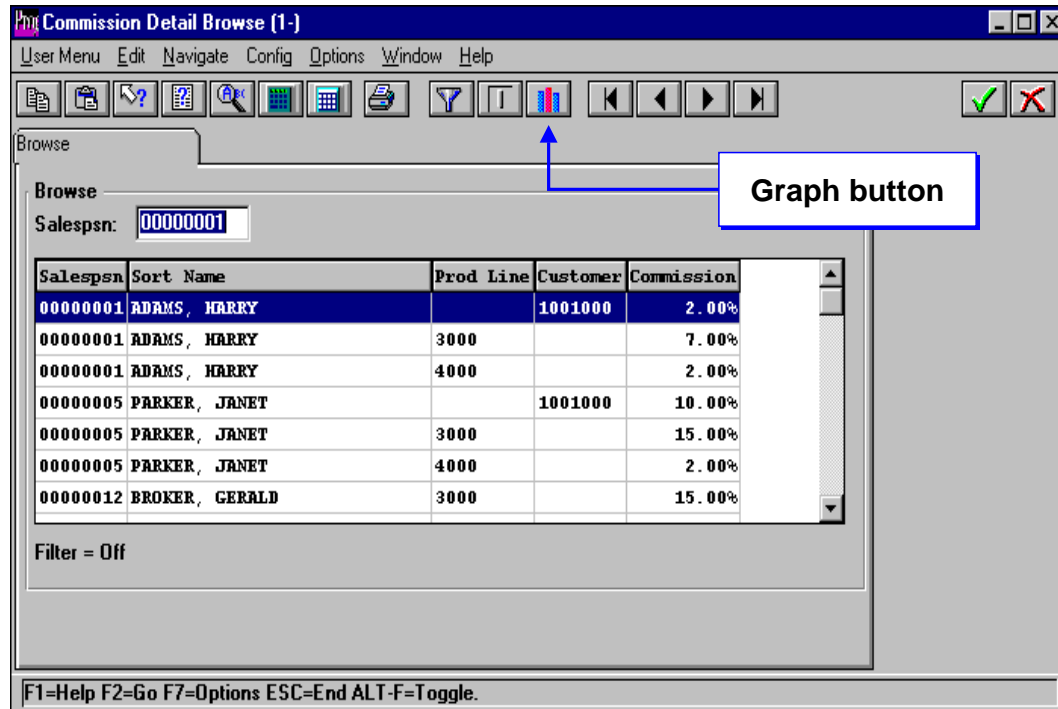
Recreate the browse from Exercise 2 in the character environment. Call it xx125 (since xx123 and xx124 are taken).

Additional Information

Example Uses Of A Power Browse

The following is a walkthrough of examples of the uses of a Power Browse. For more detailed information on Power Browse Features, please refer to the MFG/PRO User Interface Guide.

2.5.7 -- Commission Detail Browse (1-)

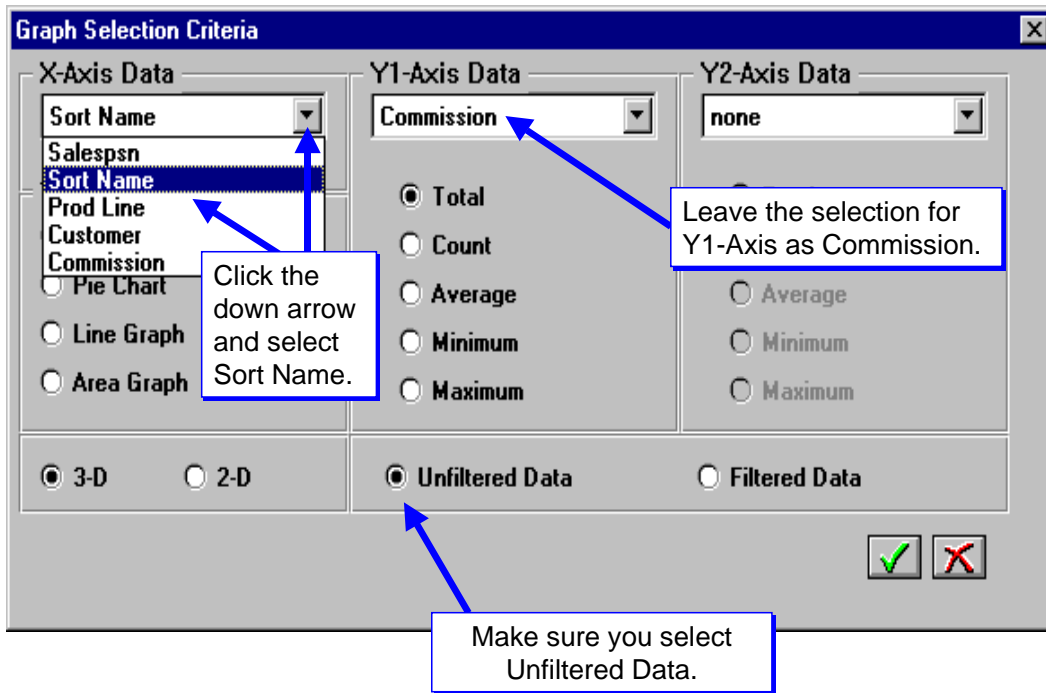


Browse Graphing

You can make graphs of browse data on the screen and print them. Browse graphing is available only in MFG/PRO for Windows. Let's walk through an example by launching the Sales Person Commission Detail Browse (adbr011.p - 2.5.7). The above display appears.

Let us graph Salesperson Commission by "Sort Name." To do this, click on the graph icon shown above.

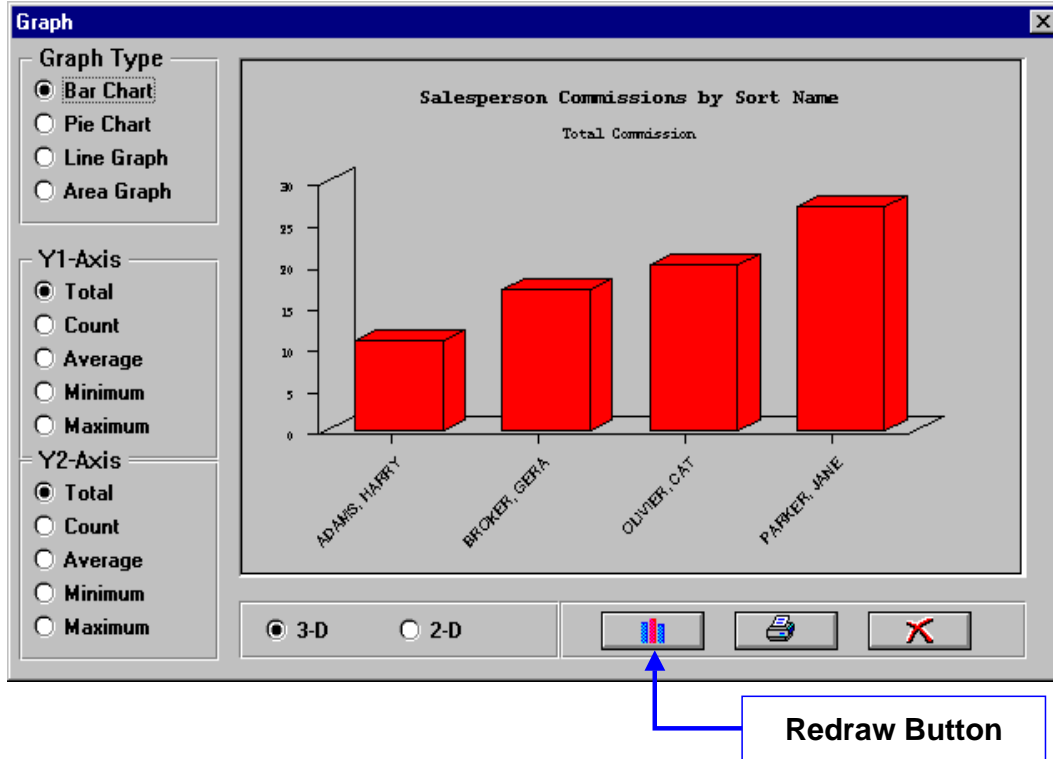
Graph Selection Criteria



The above screen appears.

Change the value of the X-Axis Data from Salespsn to Sort Name.

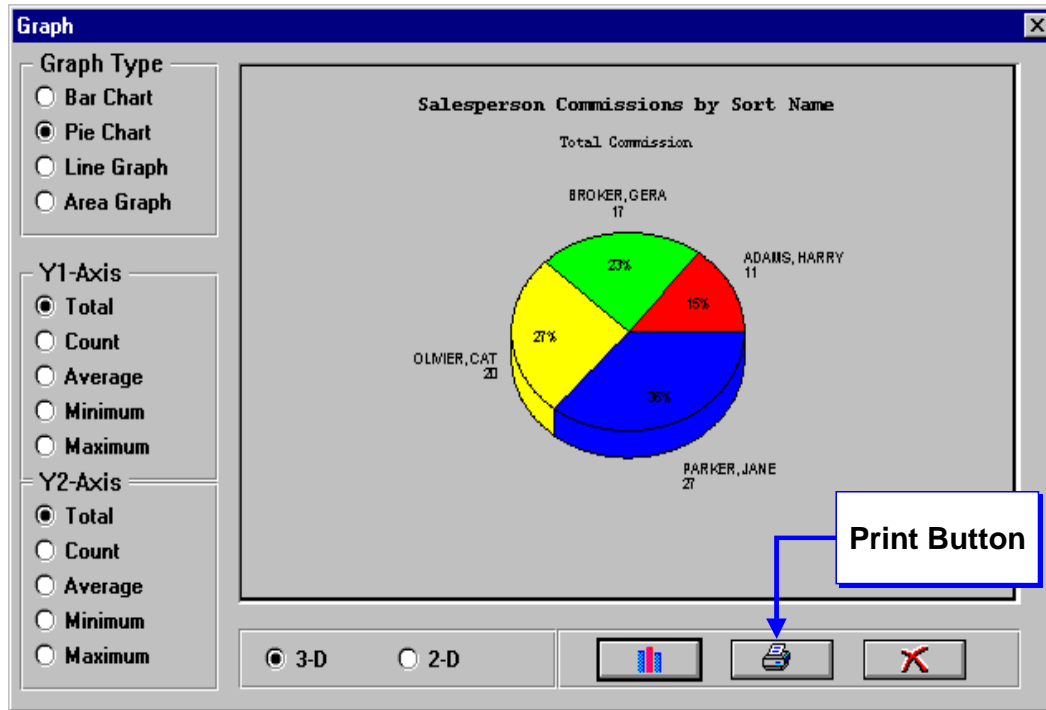
Graph



Click on the green checkmark and the above graph appears.

There are four different graph types available. The figure above shows the Bar Chart. If you wish to see the Pie Chart, select Pie Chart and then click on the redraw button.

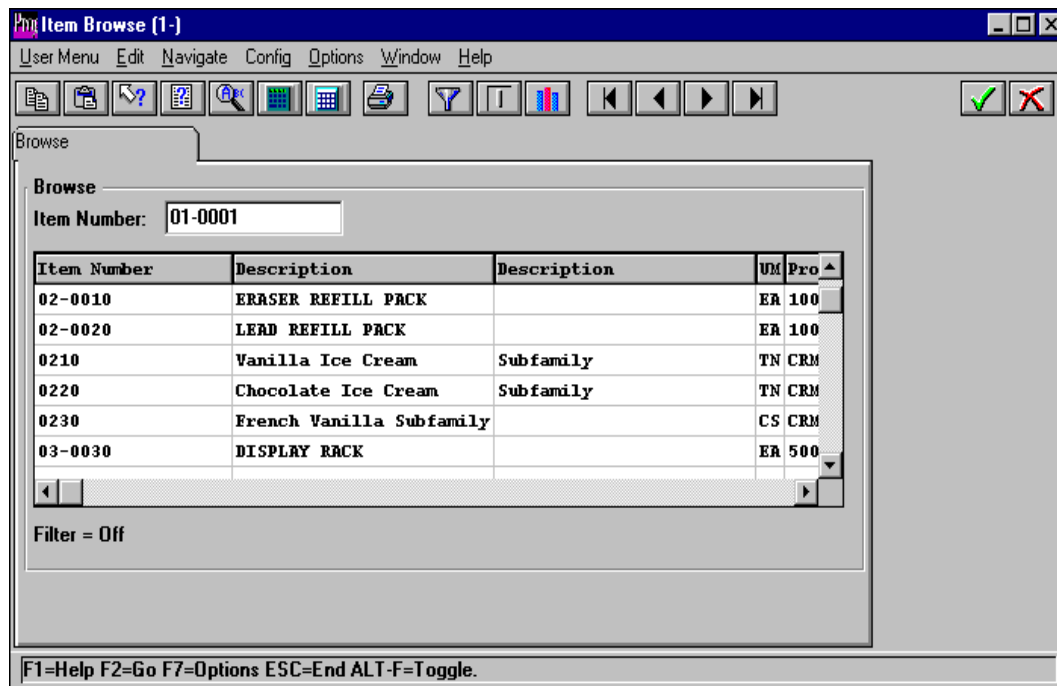
Graph



You then get the above graph.

Additionally, you can print this graph by clicking on the print button.

1.6 -- Item Browse (1-)



Filtering and Sorting Browses

Power browses have a Browse Options button on the tool bar for accessing filter and sort options. Filters limit the viewable records according to your specification. To see an example of this, run the Item Master Browse (ppbr013.p - 1.6).

Browse Options

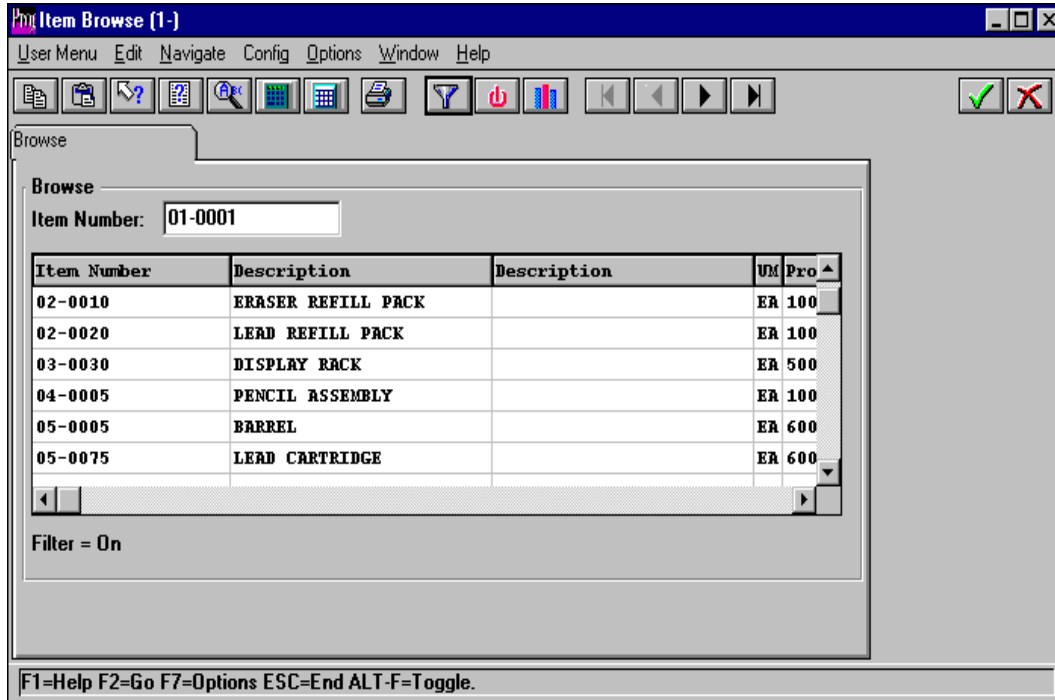
Item Number:		To	*
Description:		To	*
Description:		To	*
UM:	EA	To	EA
Prod Line:		To	*
Rev:		To	*
Drawing:		To	*
Item Type:		To	*
Status:		To	*
Group:		To	*

Sort By:

Sort Order:
 Ascending
 Descending

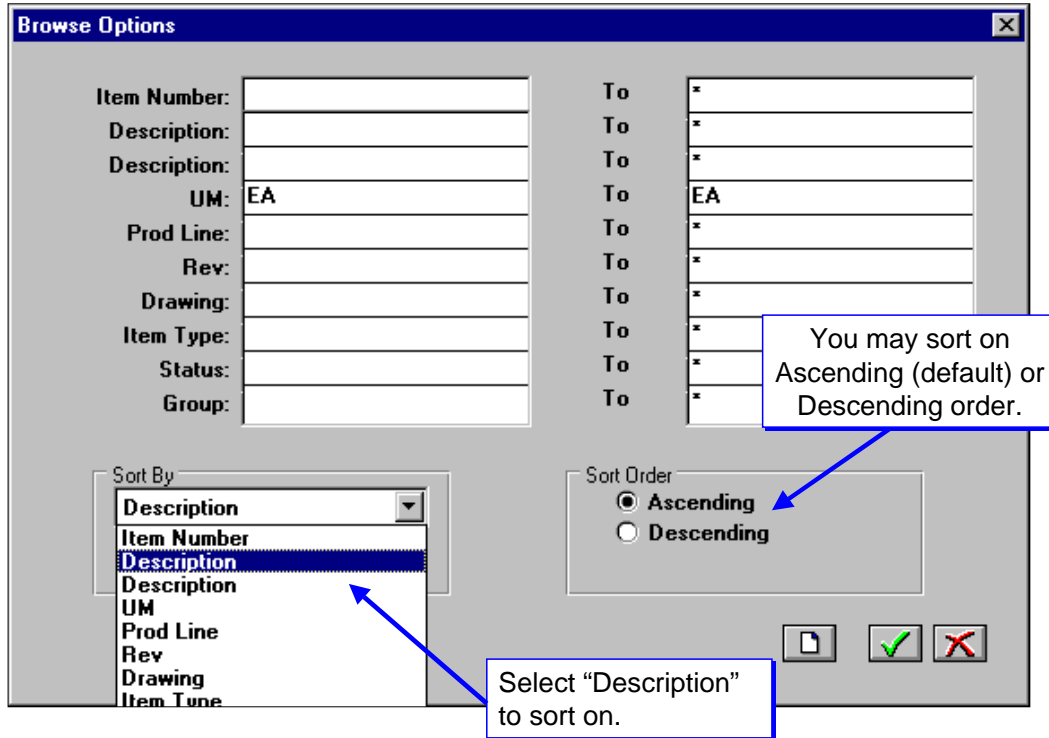
Suppose you only want to display items that have a UM of "EA". This can be done by clicking on the filter button. MFG/PRO displays the Browse Options window. In here, you specify "EA" in the "from" and "to" field for UM. See figure above.

1.6 -- Item Browse (1-)



Click on the green checkmark to proceed and you see only the items the have a UM of "EA".

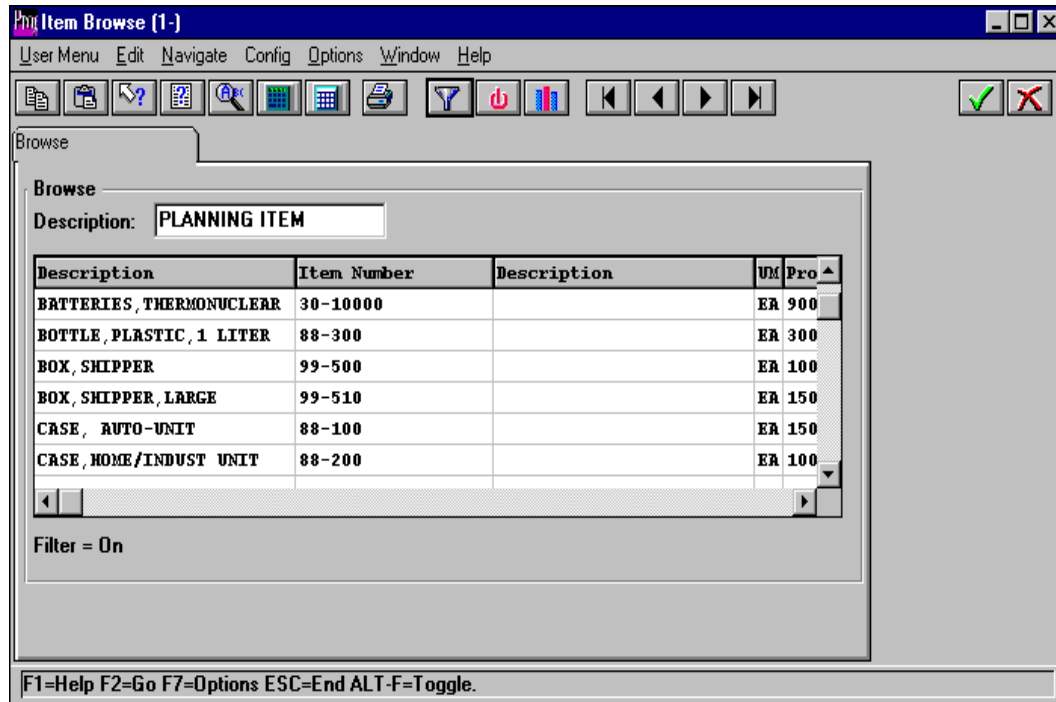
Browse Options



You can also use the Browse Options window to sort your results. Sort the results by the Item Description. To do this, click on the filter button again. The above window appears again.

Change the "Sort By" option to Description. Then click on the green check to display your results.

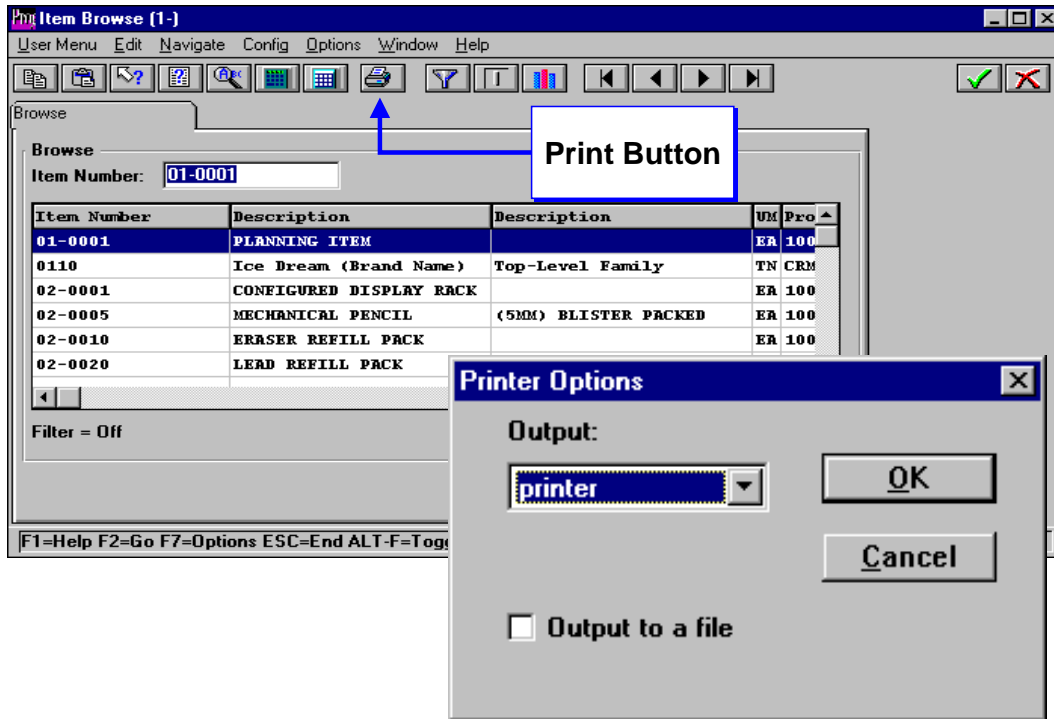
1.6 -- Item Browse (1-)



Here you see the results of your sort by description in ascending order.

Please refer to pages 60-61 of the MFG/PRO User Interface Guide for more information on Filtering and Sorting Browsers.

1.6 -- Item Browse (1-)

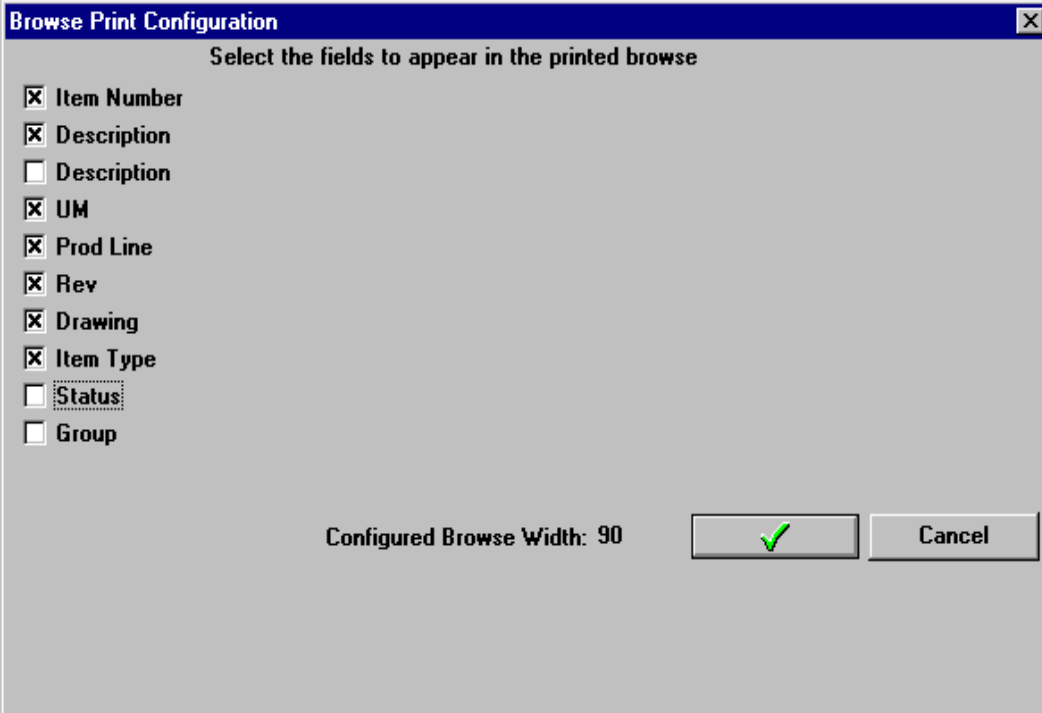


Printing your Browse

To print the results of our example Item Master Browse, click on the print button.

You see the Printer Options displayed. Click OK to print to the printer.

Browse Print Configuration



The screenshot shows a dialog box titled "Browse Print Configuration" with a close button (X) in the top right corner. The main text inside the dialog reads "Select the fields to appear in the printed browse". Below this text is a list of fields, each with a checkbox:

- Item Number
- Description
- Description
- UM
- Prod Line
- Rev
- Drawing
- Item Type
- Status
- Group

At the bottom of the dialog, there is a label "Configured Browse Width: 90" and two buttons: a green checkmark button and a "Cancel" button.

A Browse Print Configuration screen appears. Here you can select the fields you want to appear in the report. In this example we omit printing the second Description, Status, and Group.

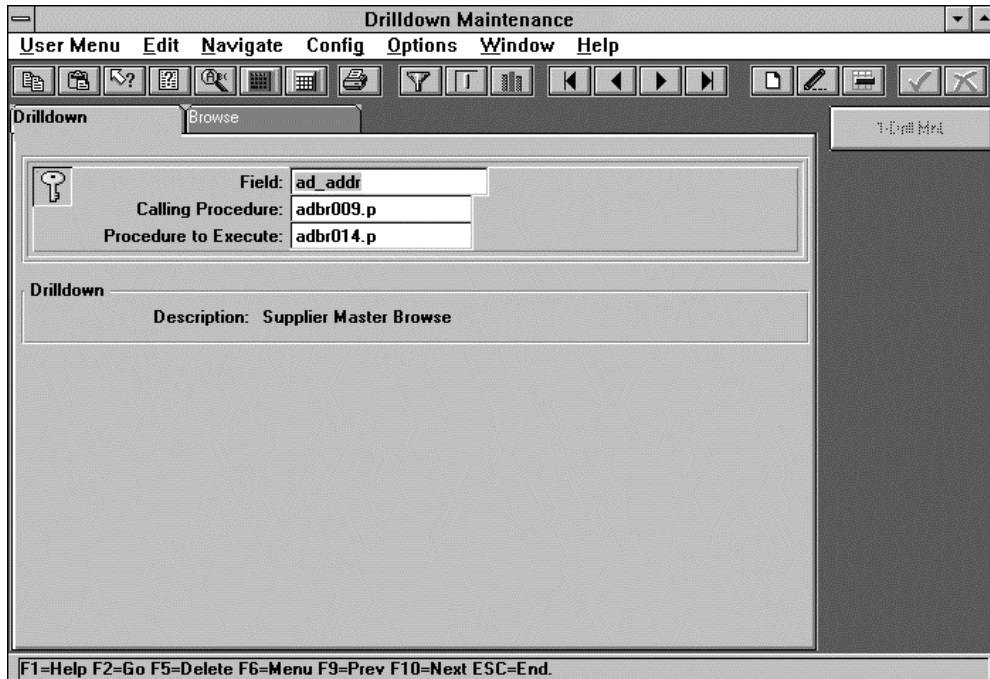
Click on the green check to continue.

Sample Report

ppbr013.p		1.6 Item Browse		Date: 07/30/98		
Page: 1		MFG/PRO Training DB - V85		Time: 10:27:42		
Item Number	Description	UM	Prod Line	Rev	Drawing	Item Type
01-0001	PLANNING ITEM	EA	1000			Family
0110	Ice Dream (Brand Name)	TN	CRMS			
02-0001	CONFIGURED DISPLAY RACK	EA	1000			CONFIG
02-0005	MECHANICAL PENCIL	EA	1000			FINGOOD
02-0010	ERASER REFILL PACK	EA	1000			FINGOOD
02-0020	LEAD REFILL PACK	EA	1000			FINGOOD
0210	Vanilla Ice Cream	TN	CRMS			
0220	Chocolate Ice Cream	TN	CRMS			
0230	French Vanilla Subfamily	CS	CRMS			
03-0030	DISPLAY RACK	EA	5000		D	FINGOOD
04-0005	PENCIL ASSEMBLY	EA	1000	1	702	FAB
0411	Vanilla Ice Cream	CS	CRMS			
0412	Vanilla Ice Cream	CS	CRMS			
0421	Chocolate Ice Cream	CS	CRMS			
0422	Chocolate Ice Cream	CS	CRMS			

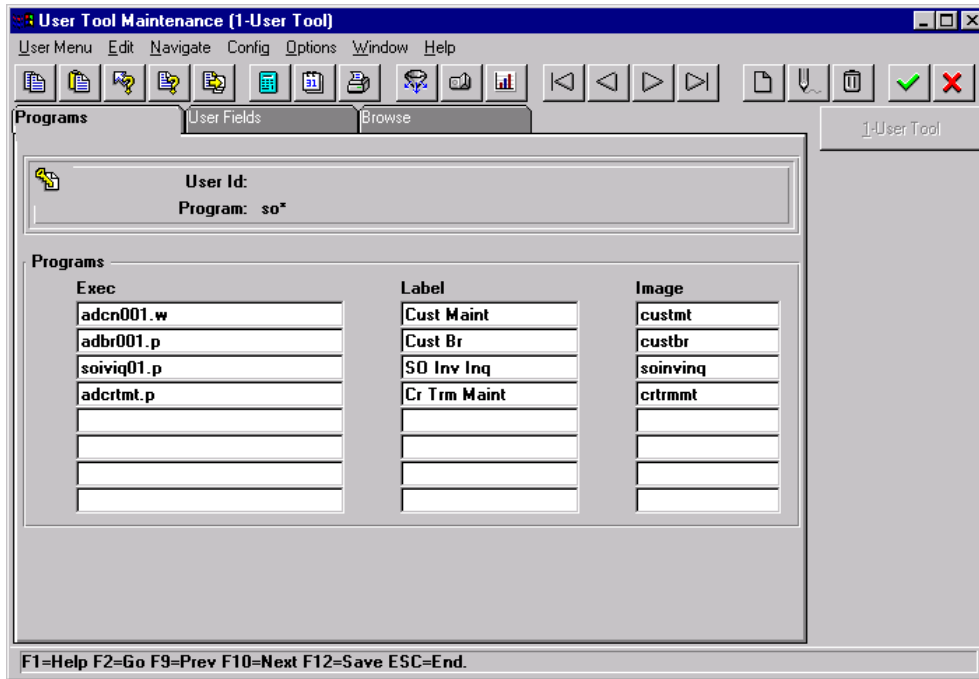
The report goes to the printer that MFG/PRO is set to.

36.20.1 -- Drilldown Maintenance



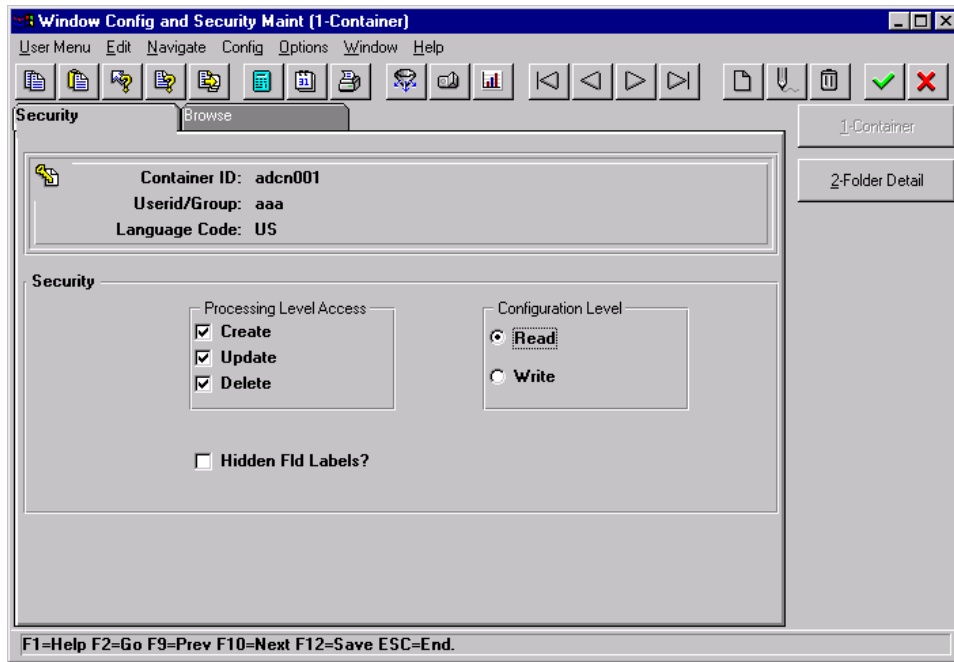
Drilldown Maintenance

36.20.2 -- User Tool Maintenance (1-User Tool) - Programs Folder



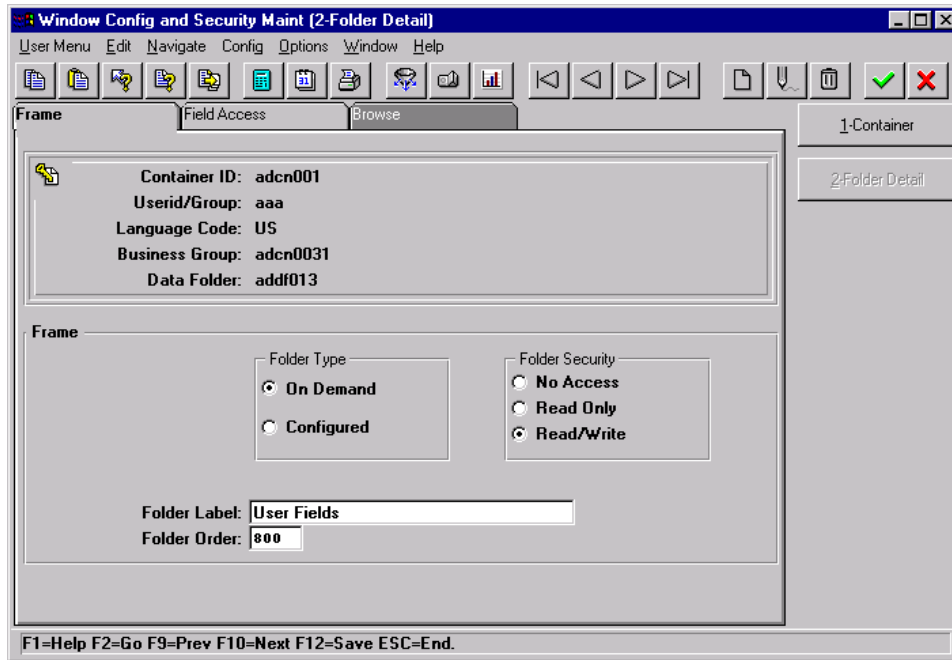
User Tool Maintenance

36.20.5 -- Window Config and Security Maintenance - Security Folder

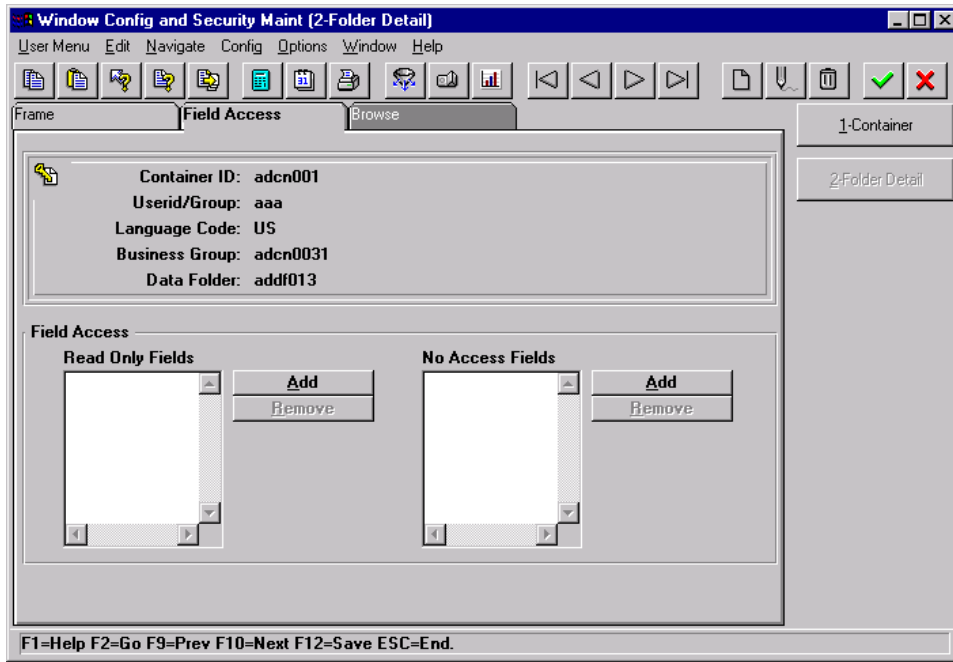


Window Config and Security Maintenance

36.20.5 -- Window Config and Security Maintenance - Security Folder, 2-Folder Detail



36.20.5 -- Window Config and Security Maintenance - Security Folder, 2-Folder Detail (Field Access)



Your Feedback Is Important To Us



Please take a few minutes to complete and return this feedback sheet. It will help us to improve the quality of our training materials. For your convenience, you can also download this form from our web site. After you have completed the form, return it to us by fax: (856) 840-2612, or send your comments to us by email.

Web site: <http://www.qad.com/services/learn>

Email: learningservices@qad.com

Training Guide: eB System Administration Progress
Database on UNIX Server

Item number: 70-2824A

How would you rate the quality of this training guide?

Organization of information:	<input type="checkbox"/> Excellent	<input type="checkbox"/> Good	<input type="checkbox"/> Fair
Ease of use:	<input type="checkbox"/> Excellent	<input type="checkbox"/> Good	<input type="checkbox"/> Fair
Content:	<input type="checkbox"/> Excellent	<input type="checkbox"/> Good	<input type="checkbox"/> Fair
Overall effectiveness:	<input type="checkbox"/> Excellent	<input type="checkbox"/> Good	<input type="checkbox"/> Fair

What suggestions, corrections, and additions to the training materials do you have?
(For specific changes, please refer to the page number and paragraph so that we can identify it.)

To order training materials from QAD, visit our web site and download the Training Materials Order Form. Send the completed order form to:

Fax: (856) 840-2612, or
email: COPS@qad.com

Your feedback is important to us. Thank you for taking the time to send it to us.



