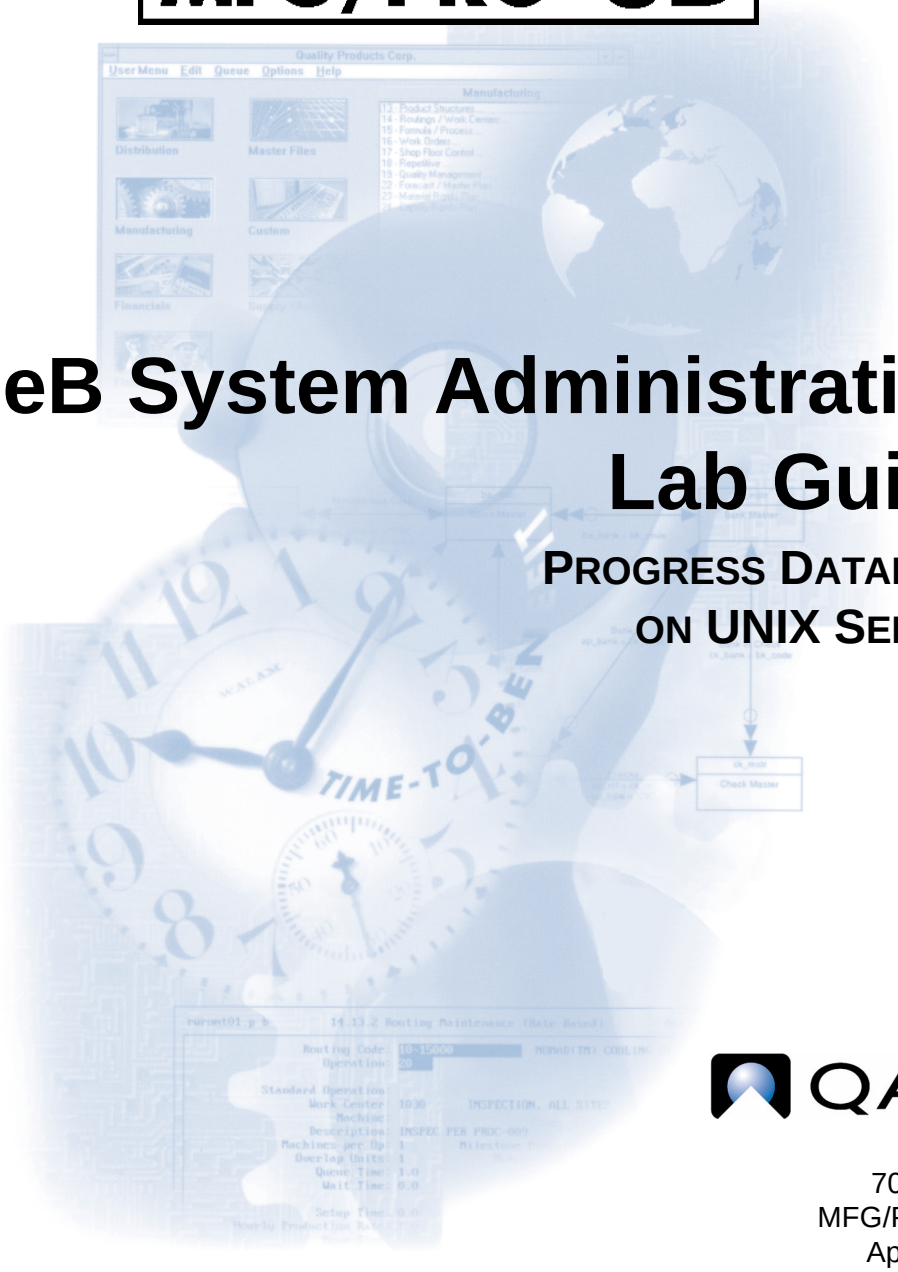
MFG/PRO® eB



eB System Administration Lab Guide

PROGRESS DATABASE ON UNIX SERVER

QAD

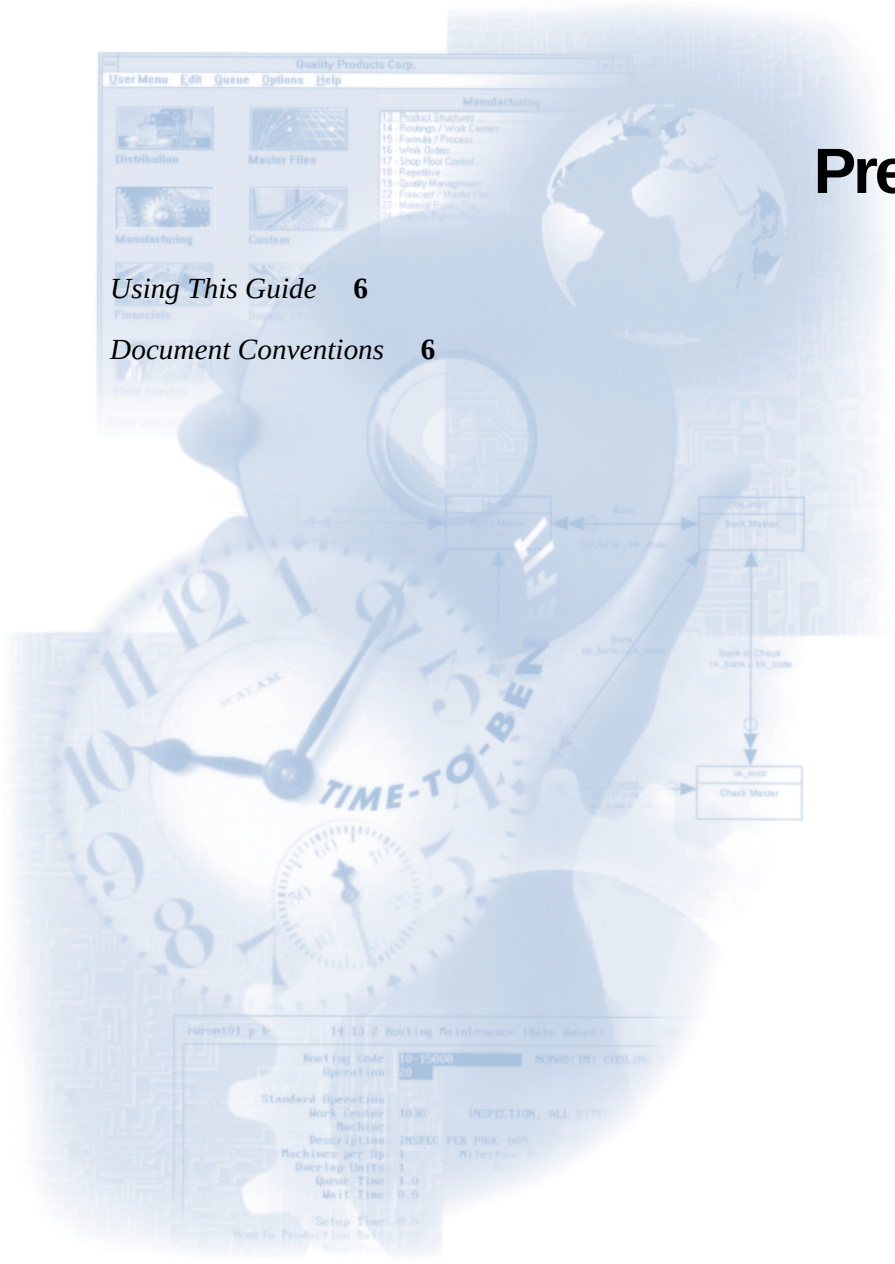
70-2825A
MFG/PRO eB
April 2001

70-2825A

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

Using This Guide

Use this guide to perform MFG/PRO system administration training labs.

Audience

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

Document Conventions

This guide covers 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, *`YourCDROMDir`*.
`indented command line`	A long command that you enter as one line (although it appears in the text as two lines).

CHAPTER 1

System Administration Labs

Lab 1: Dumping Through PROGRESS Data Administration Tool

In this activity, you will use the PROGRESS Data Administration tool to dump several tables.

1. While in a character MFG/PRO session, start the PROGRESS Procedure Editor.
   a. From the MFG/PRO Main Menu, press End.
   b. Type P at the exit confirmation prompt.
2. In the Procedure Editor, type `dict` and press RUN.
3. At the PROGRESS Data Dictionary screen, select Admin and choose Dump Data and Definitions.
4. From Dump Data and Definitions, choose Table Contents.
5. From the list of tables, select `mnd_det` through `mnt_det` and `msg_mstr`, then choose OK.
6. Select OK at the output file/Code Page screen, or alter the output file to include the desired path.

   When the process is complete, you can find the data files in the dump directory and view them using any text editor.
7. Exit Data Administration, but remain in the Procedure Editor for the next activity.

Lab 2: Preparing the Training Database for a Load

To simulate an empty database, follow these steps to delete the standard menu and message records from your training database.

Warning The following procedure is only applicable to this training activity. Do not perform these steps in any other environment.

1 In the PROGRESS Procedure Editor, type the following program to delete the menu and message records.

```
for each mnd_det:
delete mnd_det.
end.
for each mnt_det:
delete mnt_det.
end.
for each msg_mstr:
delete msg_mstr.
end.
```

2 Choose RUN to execute the program.

3 Clear the Procedure Editor of its present program, and type:

```
run mf.p
```

4 Choose RUN.

5 Log into MFG/PRO.

Note that the menus and messages are no longer present.

6 To return to the Procedure Editor. simply exit MFG/PRO

7 Remain in the Procedure Editor for the next activity.

Lab 3: Loading Through PROGRESS Data Administration Tool

In this activity, you will use the PROGRESS Data Administration tool to perform a database load of the dumped menu data.

Prerequisites: This activity assumes you have completed the following.

- "Lab 1: Dumping Through PROGRESS Data Administration Tool"
- "Lab 2: Preparing the Training Database for a Load"

1 Clear the Procedure Editor of its current program.

2 Type `dict,` and choose RUN.

3 At the PROGRESS Data Dictionary screen, select Admin and choose Load Data and Definitions.

4 From Load Data and Definitions, choose Table Contents.

5 Select mnd_det and mnt_det.

6 Enter an Output Directory. Set the Acceptable Error percentage to 100, and choose OK.

7 In the Procedure Editor, type `run mf.p` and choose RUN.

8 Log into MFG/PRO.

Note that the menus have returned and that the messages are still missing. You will load the messages using the Bulk Loader. To use the Bulk Loader, you must create the Bulk Loader description file.

9 To return to the Procedure Editor, simply exit MFG/PRO. Remain in the Procedure Editor for the next activity.

Lab 4: Creating the Bulk Loader Description File

To perform the bulk load activities, create a Bulk Loader description file by completing the following activity.

1 In the Procedure Editor, type `dict` and press RUN.

2 At the PROGRESS Data Dictionary screen, select Admin and choose Create Bulk Loader Description File.

3 At the Select Tables screen, select `msg_mstr` and press OK.

4 In the Input File field, specify the directory where your file resides and the `msg_mstr.fd` file name. In the Code Page field, accept the default. Press OK to continue.

5 When informed of completion, press OK.

You can find the msg_mstr.fd file in your dump directory. To review the file, open it in any text editor.

6 Exit Administration, and exit the Procedure Editor by typing “quit” and pressing RUN.

Lab 5: Bulk Loading the Data

Once you have created the Bulk Loader description file, begin the load into your prepared training database.

Prerequisites: This activity assumes you have completed the following.

- "Lab 3: Loading Through PROGRESS Data Administration Tool"
- "Lab 4: Creating the Bulk Loader Description File"

1 Close all MFG/PRO sessions.

2 If you are running in multiuser mode, shut down the MFG/PRO databases.

3 From a UNIX prompt, change directories (cd) to your training database directory.

4 Truncate the before-image file (.bi) with the following command.

```
$DLC/bin/proutil /DBDir/DBName -C truncate bi -G 0
```

Note This step is not required, because the Bulk Loader truncates the before-image file automatically. However, it is important to become familiar with this procedure.

5 Change directories (cd) to the dump directory containing `msg_mstr.d` and `mag_mstr.fd`.

6 From the dump directory, type the PROUTIL Bulk Loader command.

```
$DLC/bin/proutil /DBDir/DBName -C bulkload msg_mstr.fd
```

The system proceeds to deactivate all the indexes, which makes this type of load extremely efficient. The .d files are then loaded, as directed by the .fd file.

Lab 6: Rebuilding the Indexes

A relational database cannot function without indexes. Because indexes are not constructed at load time, you must rebuild them once you have completed the Bulk Loader process.

1 To run the index rebuild, enter the following:

```
$DLC/bin/proutil /DBDir/DBName -C idxbuild -TM 30 -TB 30
```

The `TM` parameter controls the sort speed, and the `TB` parameter controls blocking sizes.

2 At the following prompt, enter S (Some).

```
Select one of the following
    All - Rebuild all of the indexes
    Some - Rebuild only some of the indexes
```

3 When prompted, enter the table name `msg_mstr` and the index name `msg_ln`.

4 When prompted for another table name, type `!`.

5 If the data is correct, answer Yes at the confirmation prompt.

6 At the following system prompt, answer Yes.

```
Do you have enough disk space for index sorting?
```

The index sort space requirement is about 75% of the total size of all the tables you selected.

7 To verify, start an MFG/PRO session and enter any screen. Then exit MFG/PRO completely. A valid exit message will display.

Lab 7: Setting Up a Multi-Volume Database

Preliminary Setup

- Select Exit to the Operating System (36.24.1), and press Go.
- Verify that you are in your working directory.
- Create 4 sub-directories with the following names:

mvdb1	mvbd3
mvbd2	mvbi

1 In a text editor, create a structure description file that meets the following requirements.

Refer to the following excerpts from a database analysis report and extent listing

```
RECORD BLOCK SUMMARY FOR AREA "QADWKFL" : 33
-------------------------------------------------------
                                        -Record Size (B)- ---Fragments--- Scatter
Table               Records       Size   Min   Max  Mean    Count Factor  Factor
PUB.qad_wkfl      3570   665.2K  141  225  190   3570  1.0   1.0
```

```
RECORD BLOCK SUMMARY FOR AREA "REFERENCE" : 13
------------------------------------------------------
                                        -Record Size (B)- ---Fragments--- Scatter
Table                Records       Size   Min   Max  Mean    Count Factor  Factor
PUB.code_mstr           1468      85.3K    25   116    59     1468    1.0     1.9
PUB.fldf_mstr             13      711.0B    47    68    54       13    1.0     3.9
PUB.flh_mstr            5023      279.7K    38    88    57     5023    1.0     1.0
PUB.lngd_det             813       69.5K    49   151    87      813    1.0     1.0
PUB.mnd_det             2771      194.0K    35    91    71     2771    1.0     1.0
PUB.mnt_det             2779      147.2K    30    68    54     2779    1.0     1.0
PUB.msg_mstr            5827      467.1K    42   106    82     5827    1.0     1.5
PUB.rqpo_ref               0        0.0B     0     0     0        0    0.0     0.0
```

```
03/28/01  06:22p       131,072 m-fgtrain_13.d1
03/28/01  06:22p     1,638,400 m-fgtrain_13.d2
.
.
03/28/01  06:22p       131,072 m-fgtrain_33.d1
03/28/01  06:22p       720,896 m-fgtrain_33.d2
```

As you can see from the database analysis report - Data Storage Area 13 consists of 8 logical files. These logical files contain a total of 1,273,344 bytes of data.

Looking at the directory listing for the 2 DSA 13 extents - we see a total of 1,769,472 bytes used on the disk.

The actual disk space required to house the data consists of approximately 30-40% overhead for schema data, control data, and unused storage due to block size versus record size.

Again for DSA 33 we see extents sizes equalling 851,968 and 681,164 bytes of reported data. Again there is approximately 30% overhead.

In this exercise - we will create multiple fixed 458Kb extents, plus a variable extent to house the data for these two areas.

To figure out the total allocated database size, add up each fixed extent size from the structure file. For our training database we use the 30 default MFG/PRO eB Data Storage Areas, all with one fixed extent -- shown with a value of 128 in the structure file.

Database Name:	mvdb*YourInitials*.db
Database Extents:	Place all extents in mvdb1 directory except for the extents for Data Storage Area 13 (Reference) - and Data Storage Area 33 (Qad_wkfl). Place extents for DSA 13 in mvdb2 and the extents for DSA 33 in mvdb3. Use 3 fixed 458k extents, plus 1 variable for DSA 13, and 1 fixed 458k extent,plus 1 variable for DSA 33.
Before-Image Extents:	One fixed-length (10% total allocated database size), and one variable-length. Place these extents in the mvbi directory.

2 Create a void multi-volume database by typing the following command at a DOS prompt. Replace the variables as appropriate (the *StFile* refers to your structure description file).

```
$DLC/bin/prostrct create /DbPath/mvdbYourInitials.db /StFilePath/StFileName
-blocksize XXXX (Use Promon to determine blocksize)
```

3 Make sure the servers are shut down for the database set you have been using.

4 Copy the main database you have been using (such as mfgtrain) to the multi-volume structure. Use the following command from DOS.

```
$DLC/bin/procopy /SourceDbPath/SourceDbName
   /TargetDbPath/TargetDbName
```

Your database connections will be reconfigured to connect to this newly configured database.

Lab 8: Reconfiguring Client Database Connections

In this activity, you will reconfigure your client database connections to access your newly created database.

1 Execute MFG/UTIL from the server installation directory.

```
cd eBInstallDir
./mfgutil
```

2 Select Configure, and choose Any Database Set.

3 Select the `mfgtrain` database, and press Edit Client.

4 In the edit client screen, type the following in the Physical field.

```
mvbdYourIntials.db
```

Note You may be instructed to change the path and networking parameters as well.

5 When prompted, press OK.

6 Select Scripts, and choose Generate Server Startup.

7 Select Training, and press OK.

8 Press Yes at all subsequent prompts.

9 When complete, choose close to exit.

10 Exit MFG/UTIL. If multiuser access is required, start up your servers at this point.

Lab 9: Compiling a Program

This activity guides you through compiling a single program as if it were a custom program.

1 Copy the file *LanguageDir*`/src/mgupiq.p` to your working directory, and rename it `xxmgupiq.p`.

Note Preface all custom programs with xx so they will reside in an xx sub-directory. This prevents these files from being overwritten.

2 Edit `xxmgupiq.p,` replacing the word `terminal` with `printer`.

By default this inquiry prints to the terminal. Your custom version will go to the printer.

3 Save the file.

4 Click on the MFG/UTIL icon.

5 From the MFG/UTIL main menu, select Program, then Compile Procedures.

6 In the Compile window, complete the fields according.

- Select *VERBOSE*
- Select *GENERATE COMPILE LIST FILE* - A new window appears. Fill in the fields as shown:

Table 1.1

Source Directory	. (Working Dir)
File	leave blank
To	leave blank
Compile List File	utcompil.wrk

- Select Generate - A new window appears
- When informed the file generated successfully - Close the window

- Close the File Genration window. When you return to the compile screen, allow *USE EXISTING COMPILE LIST FILE* to remain selected.
- Select
- Fill in the remaining fields as shown in the table below:
-
-

Compile List File	utcompil.wrk
Compile Propath	.,/eBDir,/eBDir/Chrcl/us/src (Working Dir,EB DIr,EB Dir/chrcl/us/xrc)
Language Code	US
Database set	Enter your compile database set or any other established database set. **Providing the primary database has been assigned a logical database name of qaddb. Which must be unset following the compile.**
Destination Directory	. (Your working directory)

7 Choose Compile. Accept the *CODE PAGE* defaults and verify correct selections on the Verification Screen. Upon successful completion, the .r program exists in a two-letter sub-directory under a *US* language directory in your working directory.

8 Type xxmgupiq.p on any menu line.

Does the program run?

9 Link this program to a menu item, and run it.

c This only modifies the behavior of the Server copy of the mgupiq.p program. Therefore, it only affects the behavior of those sessions calling their character programs from the database server, for example, the Telent Character sessions. Modify your propath to look in the us/xx directory under your working directory.

10 Start an MFG/PRO session.

11 Type `xxmgupiq.p` on any menu line.

Does the program run?

12 Link this program to a menu item, and run it.

This only modifies the behavior of the database server's copy of the `mgupiq.p` program. It therefore only affects the behavior of those sessions calling their programs from the data server, i.e., the character client sessions.

Lab 10: Setting Up A PROGRESS Batch Client Session

This lab assumes that you have created a Batch ID and assigned the Batch ID to an MFG/PRO process or report for submittal to the Batch Request Processor. If not, first complete the Batch Lab in the Manager Functions Training Guide.

Editing the Batch Program and Input Control File

First we must create the *Program to Run.* This is a standard progress program that is used in a Progress Batch Process. This program must specify where to read input from, and where to write output to, since a Progress Batch Process does not require terminal interaction.

Using VI, or any editing utility, create *BATRUN.P* in your working directory as follows, substituting the appropriate path names:

Statements in batrun.p Program	**Comments**
PROPATH = ".,C:\QADEB,C:\QADEB\us\xrc".	Setting the Path to search for required Progress programs
INPUT from C:\WORKDIR\bat.in.	Defines the input control file
OUTPUT to C:\WORKDIR\bat.out	Defines an output file for MFG/PRO screens. This can be very useful for debugging.
RUN C:\QADEB\MF.P	Calls the main MFG/PRO program
INPUT CLOSE	Cleanup
OUTPUT CLOSE	Cleanup

Next we must create the *INPUT* File as named in the *PROGRAM TO RUN*.

In this case we call it *BAT.IN*

File Statements	**Comments**
"Login" "Password"	When MFG/PRO starts you are prompted for a login and password.
"36.14.13"	Takes the system to Menu 36.14.13 (The Batch Processor)
"YourBatchID"	Enters the batch called "YourBatchID" to be selected to process.
.	Returns to Main Menu
.	Returns to Confirm Logout
"Y"	Logout Confirm

Interactively Invoke the PROGRESS Batch Process

1 Open a DOS prompt.

2 Enter the following:

```
$DLC/bin/prowin32 /DBDir/PrimaryDB -ld /GuiDbDir/GuiDB -ld qadgui -cprcodein
ISO8859-1 -b -p /YourWorkingDir/batrun.p
```

The process will now run. A file called `bat.out` now exists in your working directory. This file will contain all the output from the hidden MFG/PRO session.

The previous command line can be entered via a script, or actuated through a Cron Entry

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: Lab Guide Progress Database on UNIX Server

Item number: 70-2825A

How would you rate the quality of this training guide?

Organization of information:	❑ Excellent	❑ Good	❑ Fair
Ease of use:	❑ Excellent	❑ Good	❑ Fair
Content:	❑ Excellent	❑ Good	❑ Fair
Overall effectiveness:	❑ Excellent	❑ Good	❑ 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