

# MFG/PRO® eB

## eB System Administration Training Guide

PROGRESS DATABASE ON  
WINDOWS NT SERVER



70-2822A  
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.

**70-2822A**

**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

<b>PREFACE</b> .....	<b>7</b>
Using This Guide .....	8
Audience .....	8
Location of MFG/PRO Manager Functions Information .....	9
Organization .....	9
Document Conventions .....	10
<b>CHAPTER 1 COMPONENTS AND CONFIGURATION</b> .....	<b>11</b>
Windows NT Basics .....	13
Configuration Basics .....	15
Client/Server Configuration .....	18
Batch Mode Client .....	20
PROGRESS Basics .....	21
PROGRESS Executables .....	25
PROGRESS Parameters .....	26
PROGRESS Database Architecture .....	27
MFG/PRO Basics .....	31
MFG/PRO Code .....	32
MFG/PRO Empty Databases .....	34
Default Databases .....	35
Main MFG/PRO Databases .....	36
Other MFG/PRO Databases .....	37
Custom Side Databases .....	38

<b>CHAPTER 2</b>	<b>SYSTEM SIZING AND INSTALLATION</b>	<b>39</b>
	Disk Space Requirements	41
	Installation Overview	48
	PROGRESS Server Installation	49
	MFG/PRO for Windows NT Server Installation	52
	MFG/UTIL Basics	53
	Database Sets	55
	Server Icons	57
	Server Media File Types	58
	Directory Structure on the Database Server	60
	PROGRESS Client Installation	64
	MFG/PRO for Windows Installation	66
	MFG/PRO Icons	69
	Sample Parameter File	70
	Sample Hosts File	71
	Sample Services File	72
<b>CHAPTER 3</b>	<b>DATABASE MANAGEMENT</b>	<b>85</b>
	Data Movement	87
	Database Backups	89
	Delete and Archive	93
	Database Fragmentation	95
	Dump and Load	99
	Dump Procedure	101
	Load Procedure	103
	Bulk Load and Index Rebuild	105
	Special Dump/Load	108
<b>CHAPTER 4</b>	<b>PERFORMANCE TUNING</b>	<b>111</b>
	Tuning Techniques	113
	PROGRESS Parameters	114
	Configuring the BI file	117
	PROGRESS Performance Processes	118
	PROGRESS Monitor (Promon)	120

Using Promon to Set -B Parameter .....	121
<b>CHAPTER 5 MULTI-VOLUME DATABASE .....</b>	<b>123</b>
Multi-Volume Overview .....	125
Multi-Volume Components .....	126
Multi-Volume Creation Steps .....	127
Creating a Structure File .....	128
Sample Structure File .....	129
Determining Database and Extent Size .....	130
Calculating the Structure File Extent Size Number .....	131
Using Prostrct .....	132
Creating a Void Multi-Volume Database .....	133
Populating a Void Multi-Volume Database .....	134
Copying a Populated Database to Void Database .....	135
<b>CHAPTER 6 SYSTEM UTILITIES .....</b>	<b>137</b>
Database Set Administration .....	139
Compiling Programs .....	141
Batch Processing with the PROGRESS Batch Client .....	144
Using the NT Scheduler .....	145
Using AT Command .....	146
Establishing a Batch Mode Client Session .....	147
PROGRESS Batch Program .....	148
Batch Input File .....	149

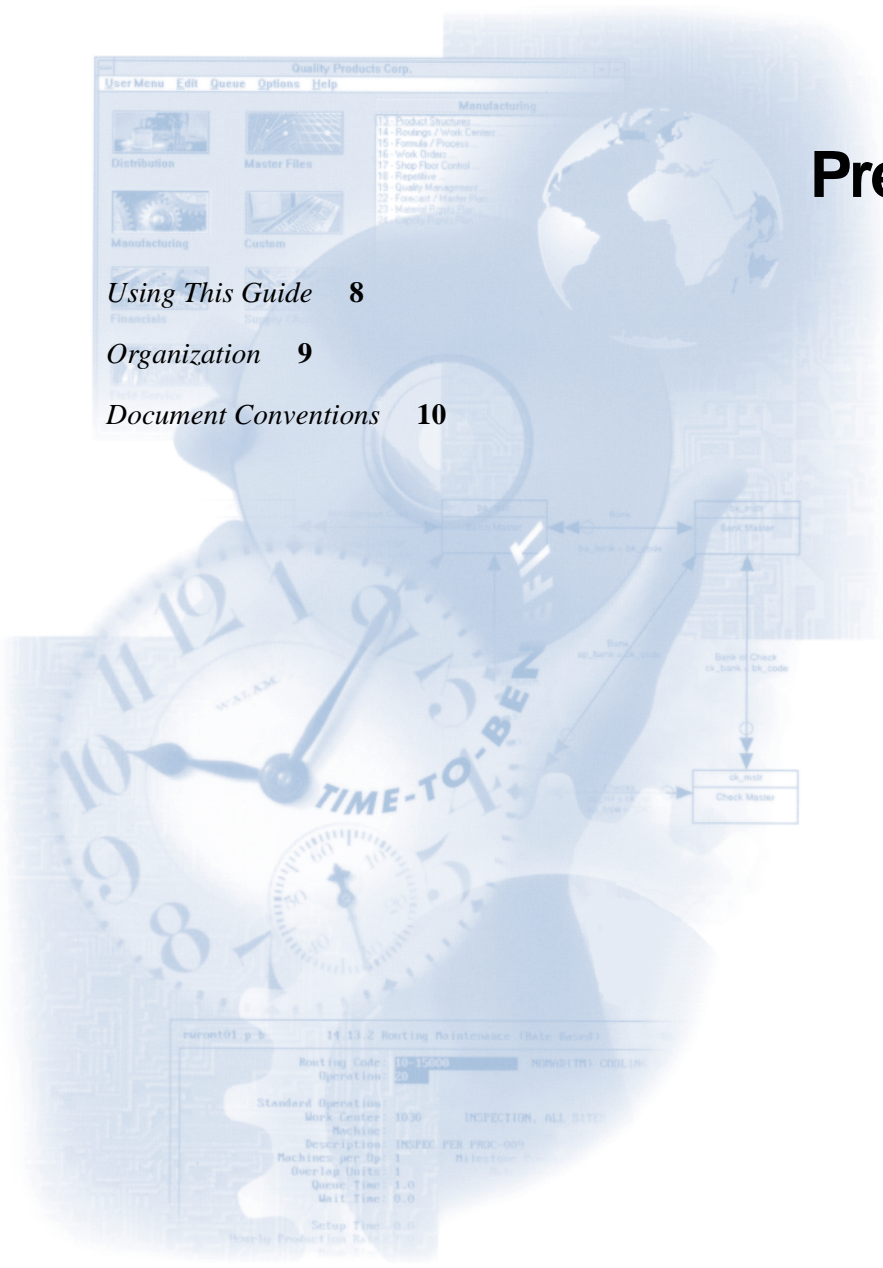
## VI EB SYSTEM ADMINISTRATION — TRAINING GUIDE

# Preface

*Using This Guide*    **8**

*Organization*    **9**

*Document Conventions*    **10**



## **System Administration for MFG/PRO on Windows NT**

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 Microsoft Windows NT and networking.

## Location of MFG/PRO Manager Functions Information

Although it is involved with system administration, this training does not include the MFG/PRO Manager Functions (menu 36). For that information, refer to *Manager 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 MFG/PRO hardware 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, which is recommended for best 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

Command prompts use the conventions listed in the following table.

<b>If you see:</b>	<b>It means:</b>
monospaced text	A command or file name.
<i>italicized</i> <i>monospaced</i> <i>text</i>	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>Windows NT Basics</i>	<b>13</b>
<i>Configuration Basics</i>	<b>15</b>
<i>Client/Server Configuration</i>	<b>18</b>
<i>Batch Mode Client</i>	<b>20</b>
<i>PROGRESS Basics</i>	<b>21</b>
<i>PROGRESS Executables</i>	<b>25</b>
<i>PROGRESS Parameters</i>	<b>26</b>
<i>PROGRESS Database Architecture</i>	<b>27</b>
<i>MFG/PRO Basics</i>	<b>31</b>
<i>MFG/PRO Code</i>	<b>32</b>
<i>MFG/PRO Empty Databases</i>	<b>34</b>
<i>Default Databases</i>	<b>35</b>
<i>Main MFG/PRO Databases</i>	<b>36</b>
<i>Other MFG/PRO Databases</i>	<b>37</b>
<i>Custom Side Databases</i>	<b>38</b>

## Components and Configuration

- Windows NT Basics
- MFG/PRO Configurations
- Progress Components
- MFG/PRO Components

## Windows NT Basics

- NT includes a DOS command line window
  - You can enter standard DOS commands
  - To create or modify text files in DOS (such as .bat scripts), use the edit command
- Instead of DOS commands, you can use Windows-based utilities
  - For example, use Notepad to edit text files
- You can set up disks and directories to be shared over the network
- To modify file attributes (such as Read Only), use the DOS attrib command or Windows Explorer

### Windows NT Basics

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

Common DOS commands appear in the following table.

**Table 1.1**  
DOS Commands

<b>Navigation Commands:</b>	
Change directory	<code>cd</code>
Change back one directory	<code>cd ..</code>
List files (long format)	<code>dir</code>
List files (a screen at a time)	<code>dir /p</code>
List files (wide format)	<code>dir /w</code>
<b>File Attribute Commands:</b>	
View attributes	<code>attrib FileName</code>
Change attributes	<code>attrib FileName +/- (A, R, etc.)</code>
<b>Other Commands:</b>	
Link (Map) a disk drive to a remote directory	<code>net use X:\\ComputerName\Dir\FileName</code> (Where X: is the drive name on the local machine.)
Open editor	<code>edit FileName</code>
Copy a file	<code>copy OldFileName NewFileName</code>
Move a file	<code>move OldDir\FileName NewDir\FileName</code>
View a file's contents	<code>type FileName</code>
View a file's contents a page at a time	<code>type FileName   more</code>
Send a file's contents to another file	<code>type FileName &gt; NewFile</code>

## Configuration Basics

- MFG/PRO can run in the following configurations
  - Host-based mode
  - Client/Server mode
- On a Windows NT database server, only client/server mode is available
- A batch client is also available, which runs in a noninteractive background host mode

### Configuration Basics

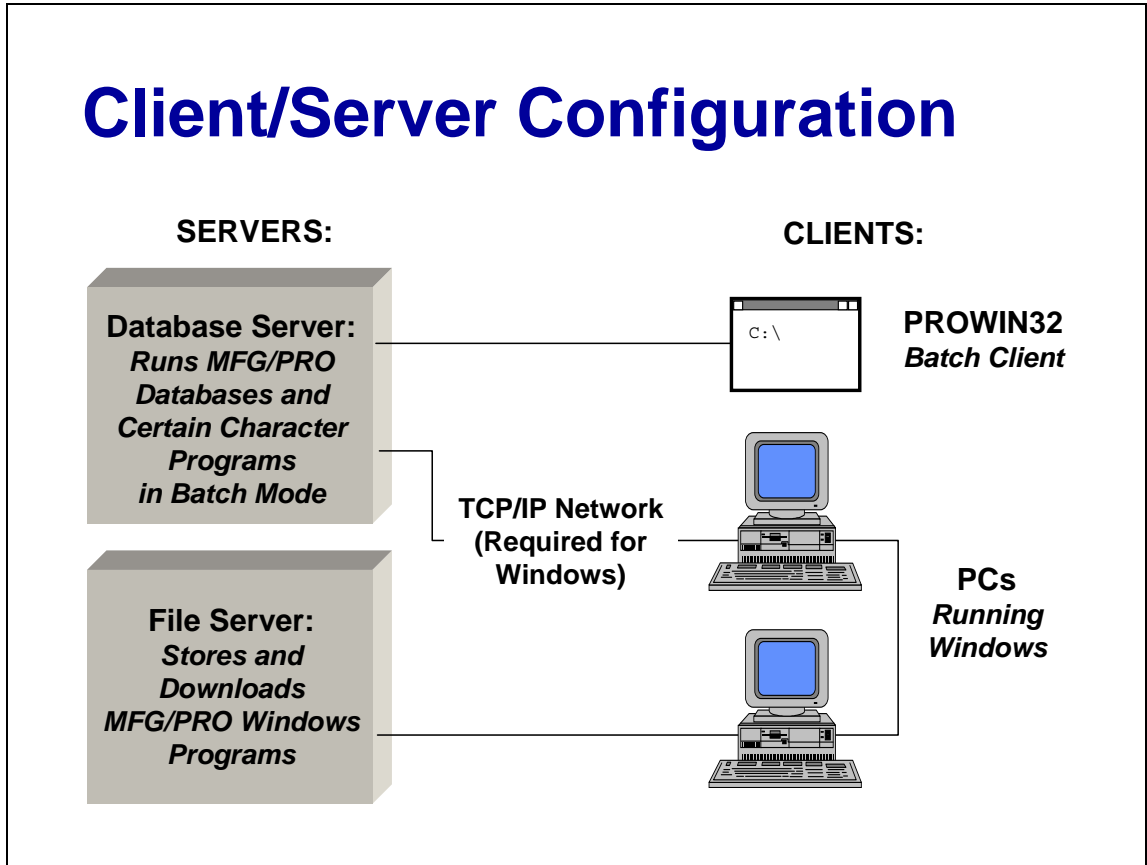
The next few slides outline configuration concepts both generally and specifically to MFG/PRO.

## 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
- ***Host mode is not supported for MFG/PRO on Windows NT***

## Client/Server Configuration

- Client/server means the client process runs on a separate processor than the database server process
  - This configuration is necessary for Windows NT
- MFG/PRO for Windows 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 *eB Installation Guide, PROGRESS on Windows NT*

## Host-Based and Client/Server Comparison

	<b>Host Client:</b>	<b>Windows Client/Server:</b>
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 January 8, 2001
  - To list installed products:  
`cat ProgressDir/progress.cfg`
- PROGRESS offers various products/packages:
  - Database server products or packages
  - Client and network connection products
  - Development and administration products or packages

config-09

### PROGRESS Basics



See *PROGRESS Basics* in the *PROGRESS System Administration Reference for 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 for 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 for 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



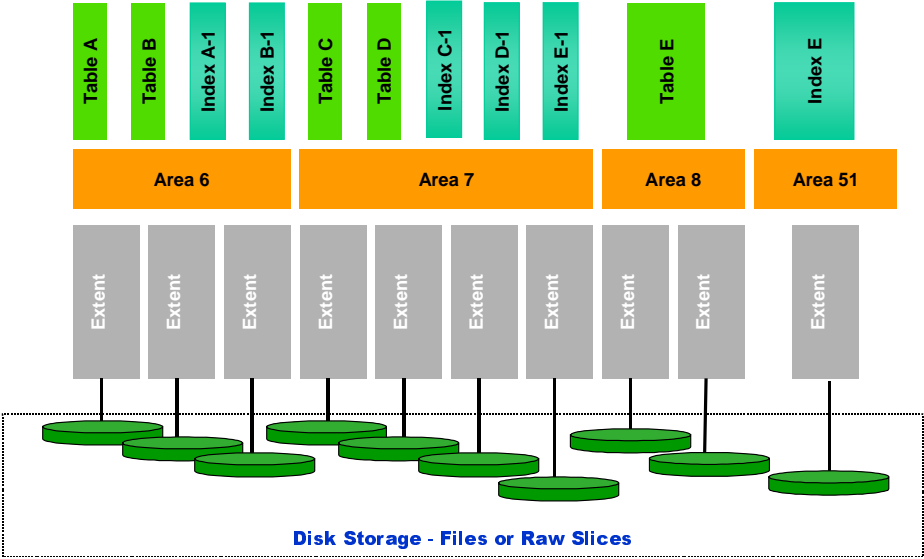
See *PROGRESS Database Architecture* in the *PROGRESS System Administration Reference for Progress Version 9*.

## **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 *Roll Backward Recovery* in the *PROGRESS System Administration Reference for 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 *eB Installation Guide, PROGRESS on Windows NT*.

## 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 *MFG/PRO Code* in the *eB Installation Guide, PROGRESS on Windows NT*.

## 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* database is void
- Empty Database
  - Copy of void database with schema file (.df) loaded
  - Empty databases are empty of data

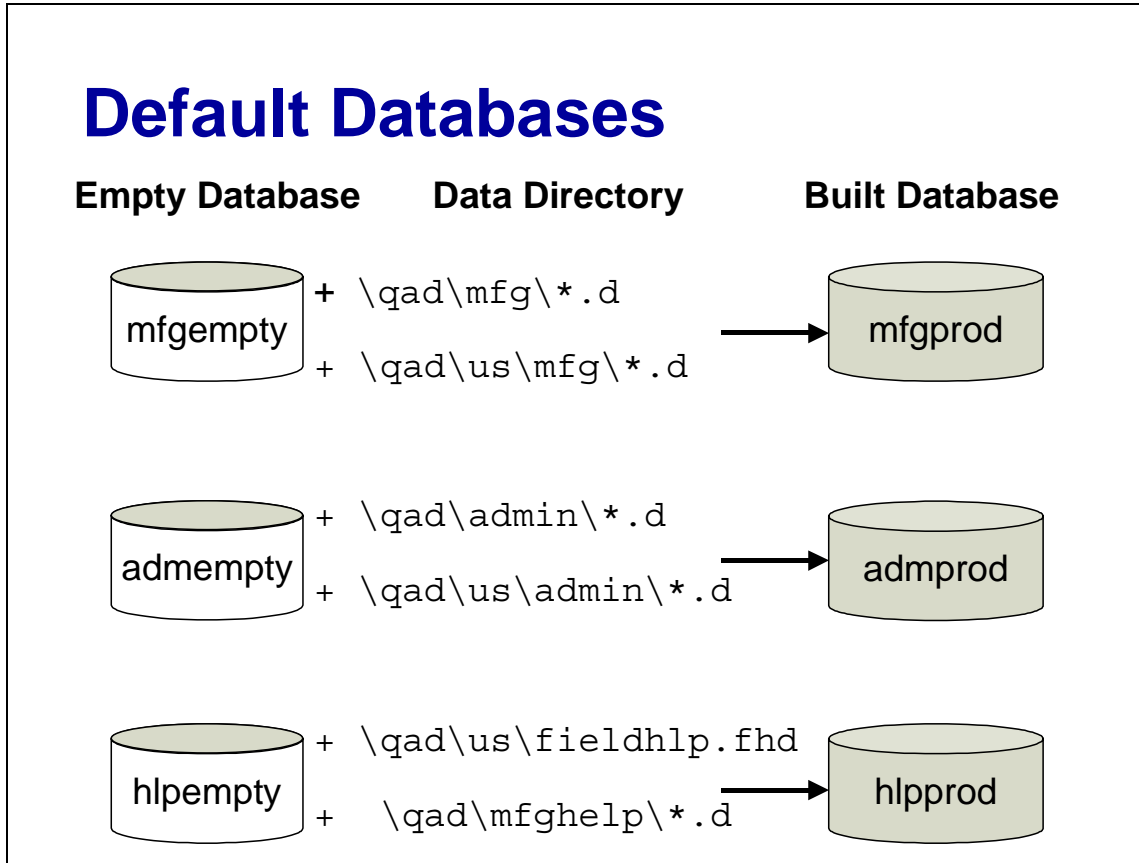
## MFG/PRO Empty Databases

- 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 *eB Installation Guide, PROGRESS on Windows NT*.



### Default Databases

- Copies of the empty databases are made and specific accumulations of predetermined data can be loaded into these copies to populate them.
- This predetermined data is stored as series of data files (.d files), located in sub-directories under the main MFG/PRO installation directory.
- Language specific files are loaded from a sub-directory beneath the language sub-directory.
- 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 complete set of .d files must be 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

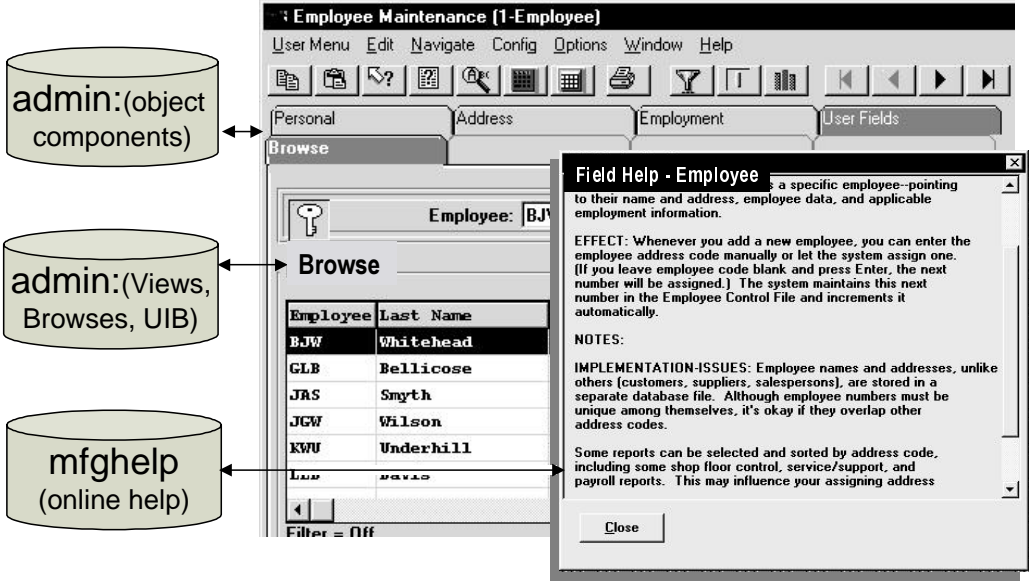


See *MFG/PRO Databases* in the *eB Installation Guide, PROGRESS on Windows NT*.

# Other MFG/PRO Databases

Database:

Related Application Item:



## Other MFG/PRO Databases



See *Support Databases* in the *eB Installation Guide, PROGRESS on Windows NT*.

## 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 *MFG/PRO Databases* in the *eB Installation Guide, PROGRESS on Windows NT*

# System Sizing and Installation

<i>Disk Space Requirements</i>	<b>41</b>
<i>Installation Overview</i>	<b>48</b>
<i>PROGRESS Server Installation</i>	<b>49</b>
<i>MFG/PRO for Windows NT Server Installation</i>	<b>52</b>
<i>MFG/UTIL Basics</i>	<b>53</b>
<i>Database Sets</i>	<b>55</b>
<i>Server Icons</i>	<b>57</b>
<i>Server Media File Types</i>	<b>58</b>
<i>Directory Structure on the Database Server</i>	<b>60</b>
<i>PROGRESS Client Installation</i>	<b>64</b>
<i>MFG/PRO for Windows Installation</i>	<b>66</b>
<i>MFG/PRO Icons</i>	<b>69</b>
<i>Sample Parameter File</i>	<b>70</b>
<i>Sample Hosts File</i>	<b>71</b>
<i>Sample Services File</i>	<b>72</b>

## 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 Requirements

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



See *Disk Space Requirement* in *MFG/PRO eB Installation Guide*, *PROGRESS on Windows NT*

## 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 *MFG/PRO eB Installation Guide, PROGRESS on Windows NT*

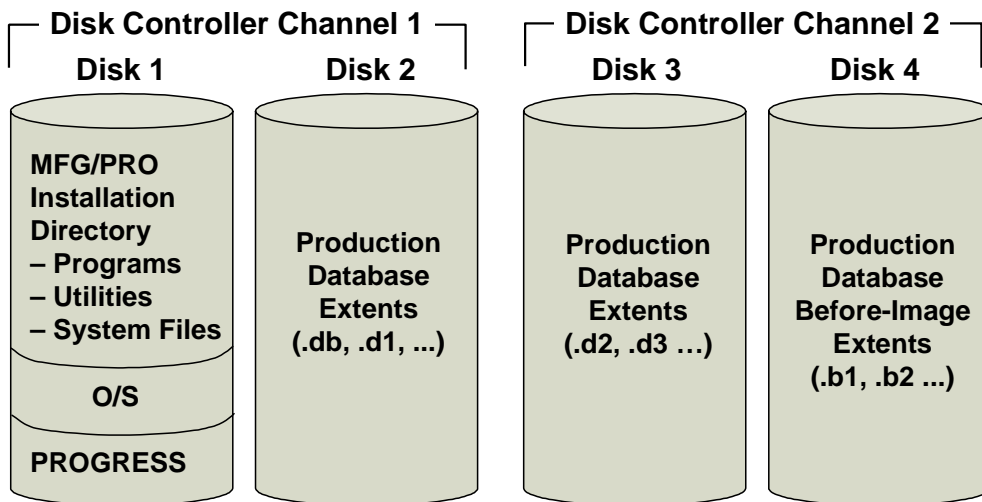
## Database Size

- 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 *MFG/PRO eB Installation Guide, PROGRESS on Windows NT*

## Disk Layout Example



- 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 *MFG/PRO eB Installation Guide*, *PROGRESS on Windows NT*

## 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 *MFG/PRO eB Installation Guide, PROGRESS on Windows NT*

## 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 *MFG/PRO eB Installation Guide, PROGRESS on Windows NT*

# 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



See *Installation Overview* and *Installation Summary* in *MFG/PRO eB Installation Guide, PROGRESS on Windows NT*

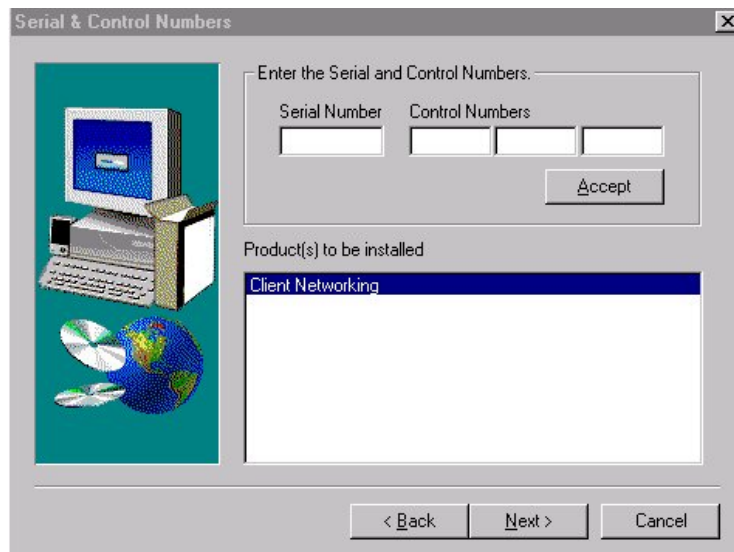
## PROGRESS Server Installation

- 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 ProVISION or 4GL license

### PROGRESS Server Installation

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

# PROGRESS Installation Window



## **MFG/PRO for Windows NT 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 Windows NT Server Installation

- Media is extracted, including database build files, language specific files, and object/source code
  - Code is character-based for use by noninteractive batch clients
  - You can also run Windows Character sessions using code on the database server; however, to avoid contention, you should install and use a second set of character-based code on a separate file server
- An InstallShield 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 for Windows NT Server Installation



See *Database Server Setup* in *MFG/PRO eB Installation Guide, PROGRESS on Windows NT*

## MFG/UTIL Basics

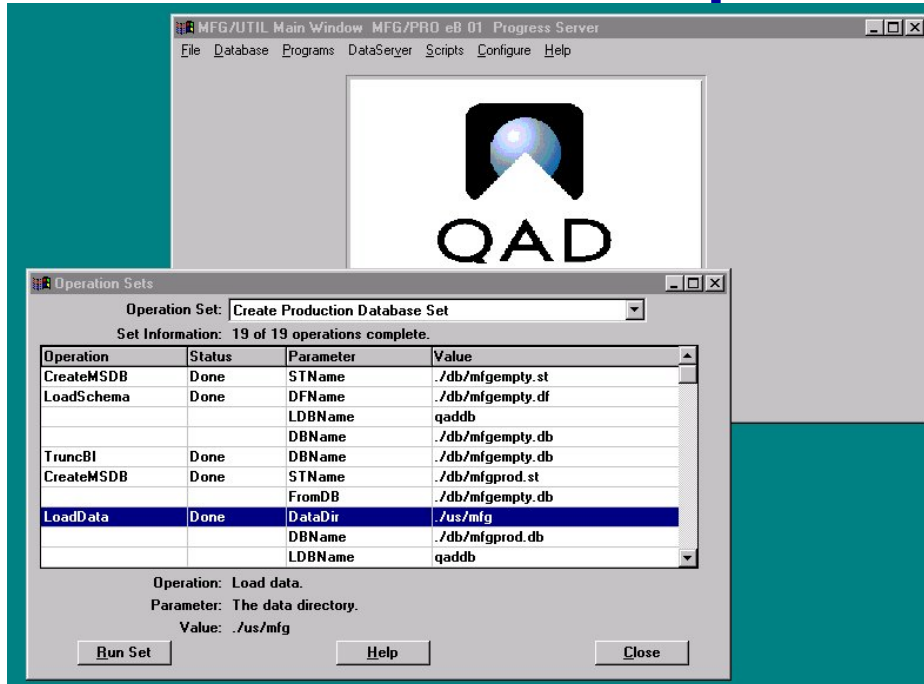
- MFG/UTIL is the installation and configuration utility
- MFG/UTIL is invoked by the install routine to:
  - Build your production databases via the “MFG/PRO Guided Setup” function
  - Configure database set specifics, including network parameters, to create start-up and shutdown icons
- After installation,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

### MFG/UTIL Basics



See *MFG/UTIL Reference Information* in *MFG/PRO eB Installation Guide, PROGRESS on Windows NT*

# MFG/UTIL Guided Setup



## Database Sets

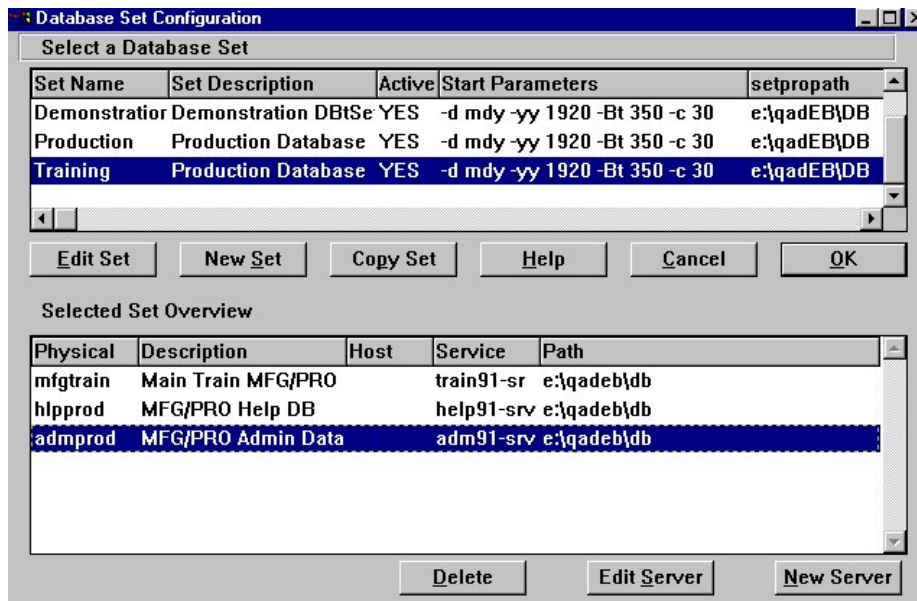
- MFG/PRO runs against a *set of databases*
  - Includes the main database, the administration database (user interface and OBCM data), the help database (on-line help), and additional/custom databases required
  - You may have database sets for different purposes, such as production or training
- MFG/UTIL 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



See *Database Sets* in *MFG/PRO eB Installation Guide, PROGRESS on Windows NT*

# MFG/UTIL Database Set Window



## Server Icons

- MFG/UTIL creates start-up and shutdown icons
- The icons run .bat files (where xxxxxxxx = the name of the database set):
  - xxxxxxxxsvst
    - Starts a server process for each database in the set
  - xxxxxxxxsvsh
    - Stops server processes for each database in the set
- As an option, you can include PROGRESS ProControl commands or AdminServer commands in the .bat files
  - NOTE: ProControl and the AdminServer is explained later in this training

### Server Icons

Take note of the server icons names on your system.

## Server Media File Types

- Delivered files:
  - .p (source code)
  - .l (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
  - .bat batch files

### Server Media File Types

**Table 2.1**  
Files Delivered with Media

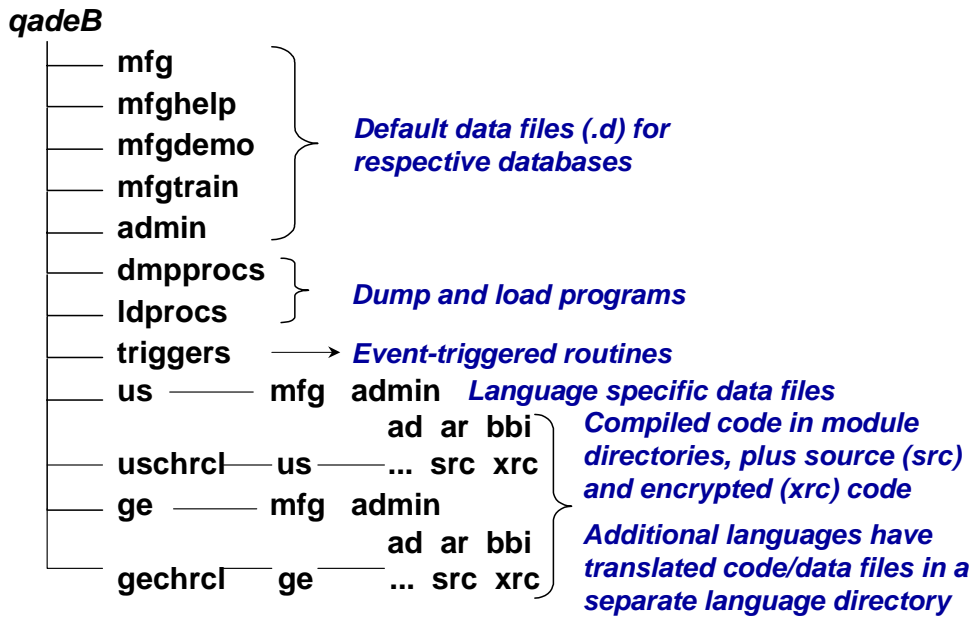
Extension	File Type	Description
.p	Source Code	Uncompiled MFG/PRO program code. The xrc sub-directory includes encrypted, uneditable source code. The src sub-directory includes any nonencrypted, editable source code you have purchased.
.i	Include	Subroutines referenced by the .p source code.
.v	Validation	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.
.fd	Bulk Load Description File	Text file used to execute a database bulk load. Contains the table names for the target database, along with the associated data files.
.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.
.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.

**Table 2.2**  
Other Database Files

Extension	File Type	Description
.bx	Database Before-Image Extent	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

## Source Code Cross-Reference

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

## 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\mfg\help* as the Input Directory
  - All Files(Tables) are selected by default – Select OK
  - Complete load
- Review `mfgutil.log` for errors

## PROGRESS Client Installation

- Same media as the Progress Server installation
- 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 disk space on each client
  - Requires multiple upgrades at upgrade time
- If you install on a file server, make sure to follow the PROGRESS network set-up instructions

### PROGRESS Client Installation

Take note of the directory for your installation of PROGRESS for Intel NT.

## **MFG/PRO for Windows Installation - Character Client**

- Character code was installed during Database Server installation
  - Contains Character object code, and optional source code
  - Does not contain database build files
  - Contains character client setup routine
- Should be re-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 Character MFG/PRO sessions

## **MFG/PRO for Windows Installation - Gui Client**

- Separate Windows Client CD
- Should be installed on a network file server
  - Contains GUI object code, and optional source code
  - Does not contain database build files
  - Contains GUI client setup routine
- 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 appropriate MFGPRO for Windows installation directory (on file server)
- Prompts for:
  - Local installation directory (for a small number of files)
  - PROGRESS and MFG/PRO code 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 *Character Client Setup* and *GUI Client Setup* in *MFG/PRO eB Installation Guide, PROGRESS on Windows NT*

## 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 NTServ -S train91-srv -N tcp
-db admprod -ld qadgui -H NTServ -S adm91-srv -N tcp
-db hlpprod -ld qadhelp -H NTServ -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 to better understand client/server connections.

## Sample Hosts File

```
# MFGPRO db hosts addresses and names
```

```
192.152.71.42    NTServ1
192.152.71.43    NTServ2
192.152.71.44    NTServ3
192.152.71.45    NTServ4
```

*IP Address*

*Host Name*

### Sample Hosts File



See *Modifying Hosts and Services Files* in *MFG/PRO eB Installation Guide, PROGRESS on Windows NT*

## 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 *MFG/PRO eB Installation Guide, PROGRESS on Windows NT*

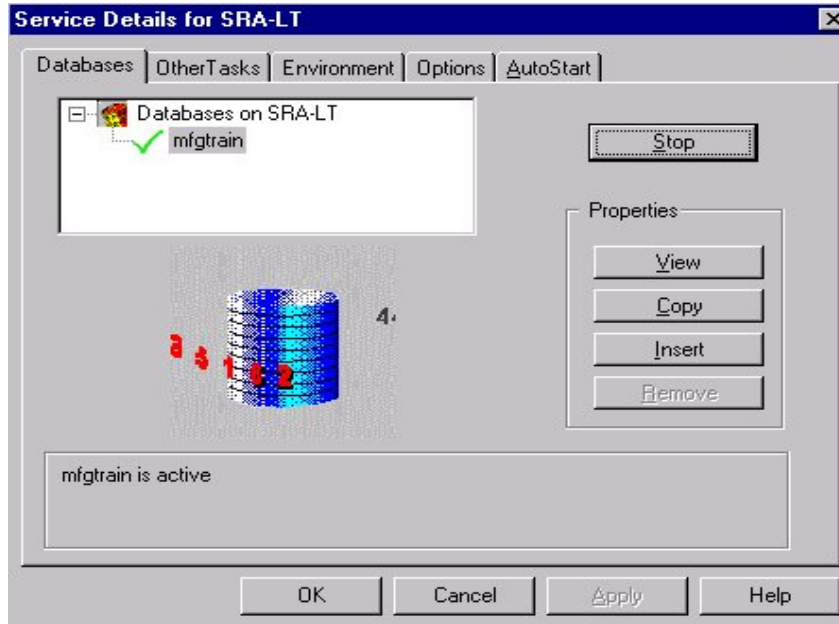
## Progress DataBase Startup

- MFG/PRO Database Server Installation produces database startup .bat files based on the database sets
  - The user that activates these .bat files “own” the database processes, and these process will end when that user logs off
- To allow the system to own these processes – they must be started as NT Services
- ProControl and the AdminServer are Progress Utilities to control the databases as the SYSTEM
- The Utilities are listed in Control Panel -- Services

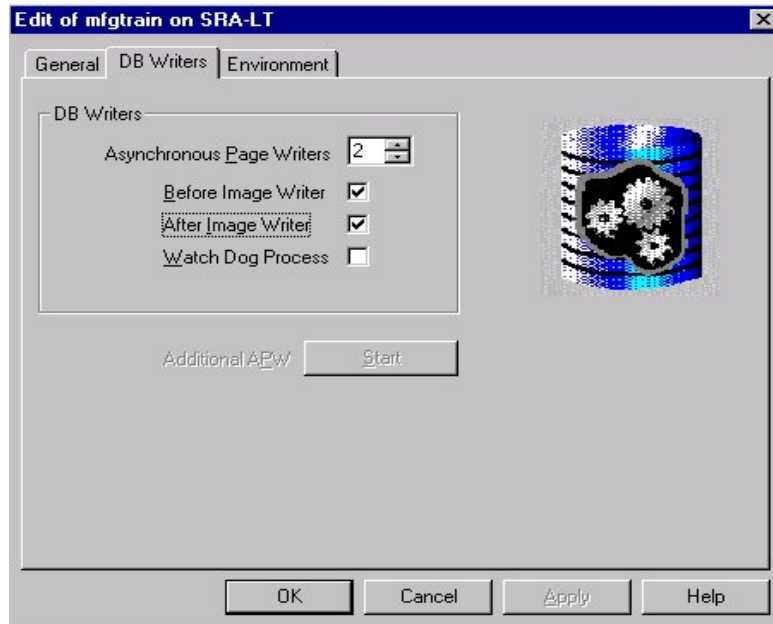
## Progress ProControl Utility



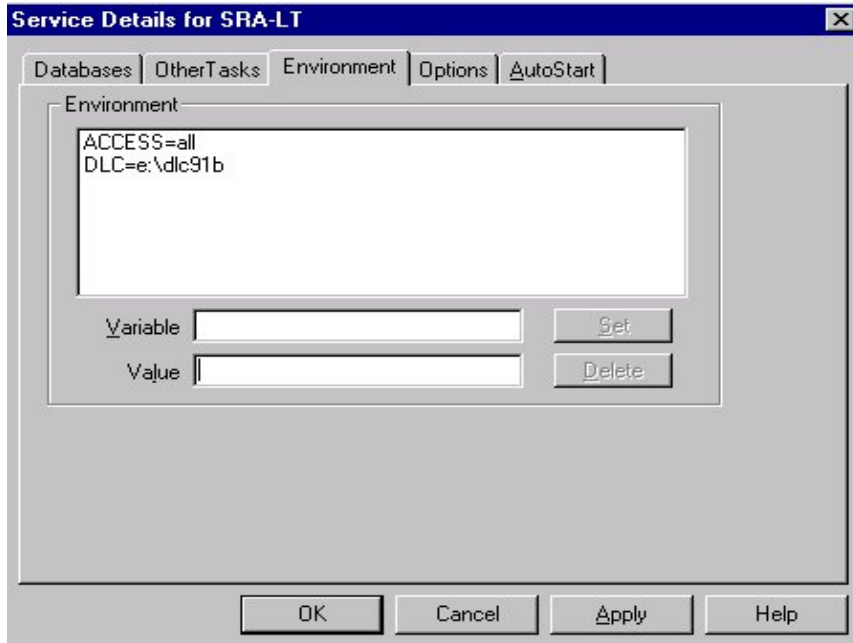
# ProControl Database Activation



## ProControl DB Writers



# ProControl Environment Settings



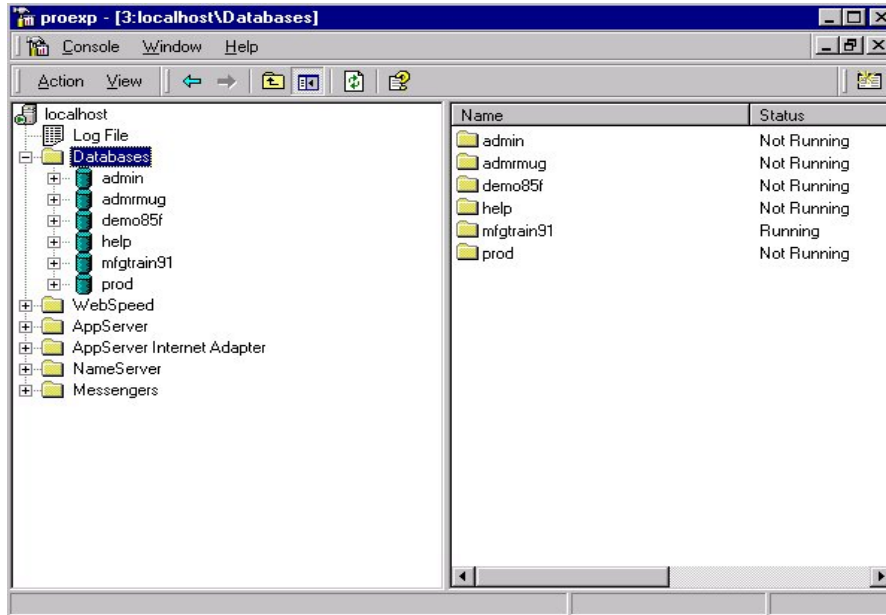
## ProControl DataBase Settings

The screenshot shows a Windows-style dialog box titled "Edit of mfgtrain on SRA-LT". It has three tabs: "General", "DB Writers", and "Environment". The "General" tab is active. The fields are as follows:

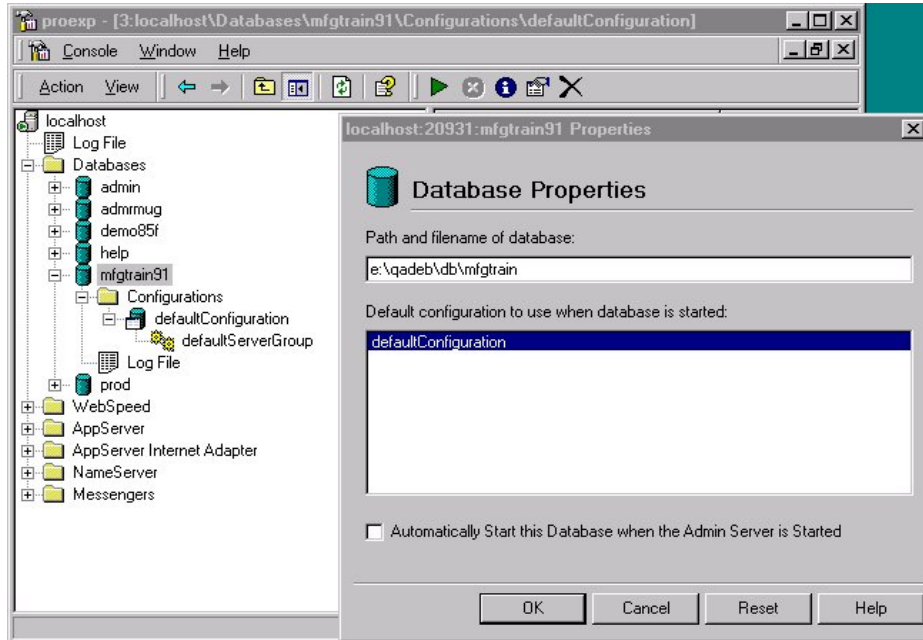
- Identifiers: mfgtrain
- AutoStart:
- Executable Name: \_mprosv.exe
- Working Directory: e:\qadeb\db
- Physical DB-Name: mfgtrain (with a "Browse..." button)
- Startup Parameters: -S train91-srv -N tcp -spin 2000 -n 80
- Shutdown Parameters: (empty)
- Priority: Normal (dropdown menu)
- Comment: (empty)

At the bottom, there are four buttons: "OK", "Cancel", "Apply", and "Help".

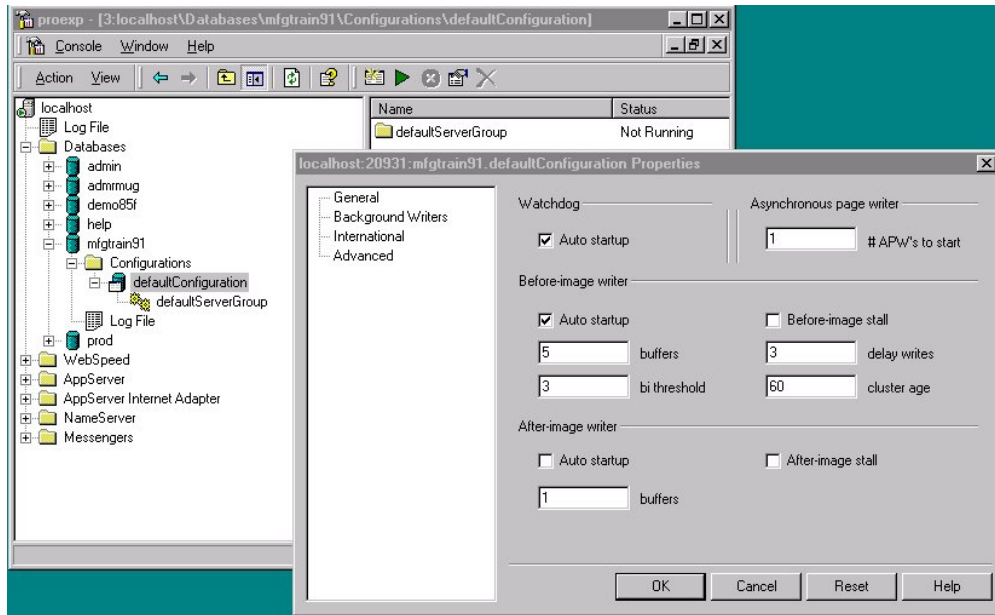
# Progress Explorer Tool



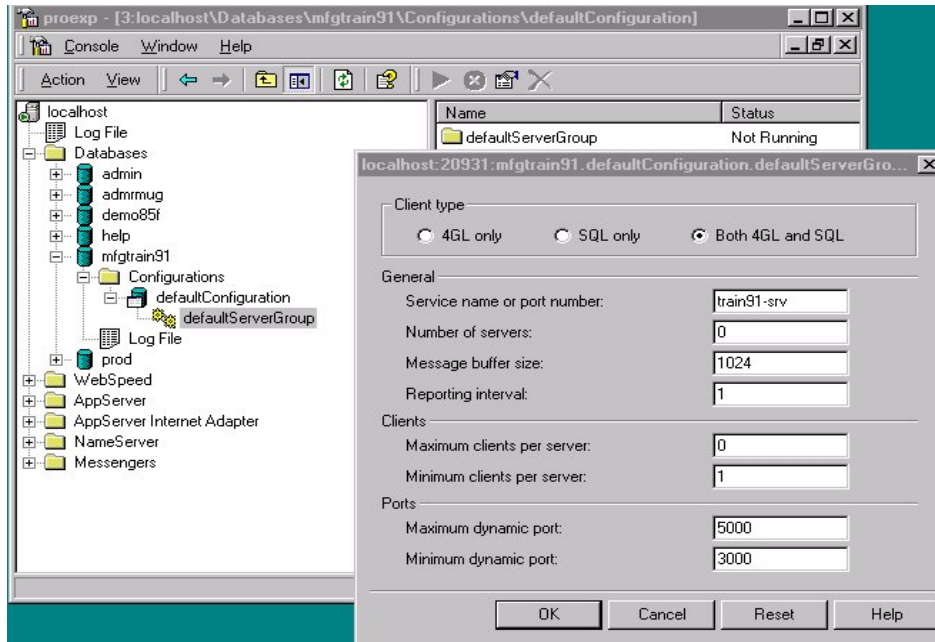
# Progress Explorer Configuration



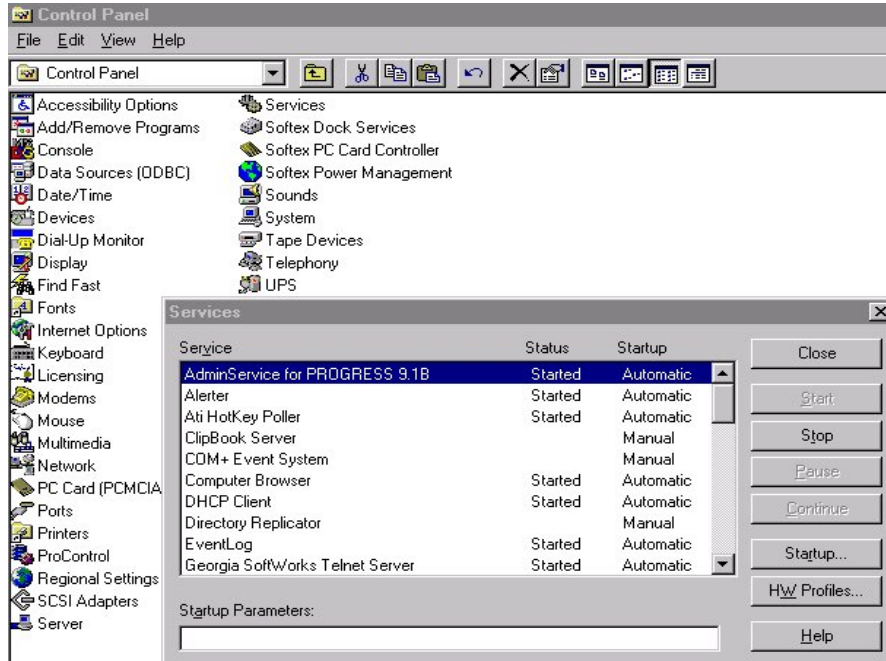
# Progress Explorer Configuration



# Progress Explorer Configuration



# NT Control Panel - Services





# Database Management

*Data Movement* **82**

*Database Backups* **89**

*Delete and Archive* **93**

*Database Fragmentation* **95**

*Dump and Load* **99**

*Dump Procedure* **101**

*Load Procedure* **103**

*Bulk Load and Index Rebuild* **105**

*Special Dump/Load* **108**



parent01 p b 14.13.2 Routing Maintenance (Date Based)

Routing Code:	10-15000	NONAUT (TR) COBL IN
Operation:	20	
Standard Operation:		
Work Center:	1030	INSPECTION, ALL SITE
Machines:	1	
Description:	INSPEC PER PROC 00%	
Machines per Op:	1	Reflection %
Overlap Units:	1	
Queue Time:	1.0	
Wait Time:	0.0	
Setup Time:	0.0	
Base to Product Line:	10-15000	

## Data Movement

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

## Data Movement

<b>Types</b>	<b>When</b>	<b>What It Does</b>	<b>Comment</b>
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
  - NT Administrative Tools Backup Exec or other Windows NT backup utility
  - PROGRESS utilities: probkup and prorest
- Delete and Archive
  - Various MFG/PRO functions
- Dump and Load
  - MFG/PRO
  - PROGRESS Data Administration
  - 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 NT 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 and integrity often

### Database Backups



See *Probkup/Prorest* and *Database Backups* in the *PROGRESS System Administration Reference, Progress Version 9*

## Backup Utilities

UTILITY	TO	WHAT	TYPE	VERIFY
NT Admin Tool	Tape	anything	Full	No
PROGRESS probkup	Tape or Disk	db and bi extents only	Incremental, Full online*	No

\* When using probkup -online option: database server and after-imaging may be running. Otherwise, they **must** be disabled.

## Backup Preparation

DO	USE	WHY
Shut down the server	xxxxxxsvsh.bat proshut	Avoid data corruption.
	Progress Utility = _mprshut	PROGRESS backup using -online option avoids corruption.
Truncate your .bi file	proutil	This eliminates data from the .bi file, 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 (NonCumulative)**
  - 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

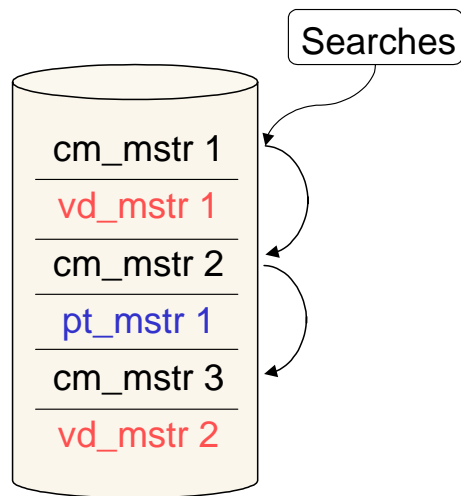
For a full explanation of delete and archive functions, refer to the *Manager Functions Training Guide*.

## 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



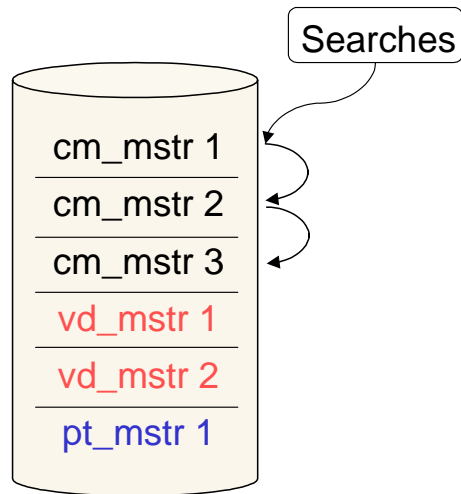
See *Database Dump and Load* in the *PROGRESS System Administration Reference, Progress Version 9*

## 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 System 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 System Administration Reference, Progress Version 9*



See *Dump and Load* in the *Manager Functions Training Guide*

## Dump Procedure (cont.)

### 3 Verify you are the only user on the system

- Use a single-user batch mode client 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

### 4 Choose one of the following methods:

- In MFG/PRO, run Database File Dump/Load w/ Batch
- Use the PROGRESS Data Administration 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\proddb NewProdDb.db mfgempty`
- 2 If you want to bulk load, create a description file
  - In Data Administration, Admin menu, choose Create Bulk Loader Description File
- 3 Verify you are the only user on the system
  - Single-user batch mode client 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 System Administration Reference, Progress Version 9*



See *Dump and Load* in the *Manager Functions Training Guide*

## 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 Administration load
  - In Data Administration, Admin menu, choose Load Data and Definitions, then Table Contents
- Use the MFG/PRO Database File Dump/Load w/ Batch
- Use MFG/UTIL

### 5 If you used the bulk loader, rebuild the indexes

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



Complete the following Labs: “Preparing the Training Database for Load,” 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 Bulkload* and *Bulkload Descriptor File* in the *PROGRESS System Administration Reference, Progress Version 9*



Complete the following Labs: “Creating the Bulk Loader Description File,” “Bulk Loading 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 Space Dump and Load* in the *PROGRESS System 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 36.16.4 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 Space Dump and Load* in the *PROGRESS System Administration Reference, Progress Version 9*

# Performance Tuning

*Tuning Techniques* 113

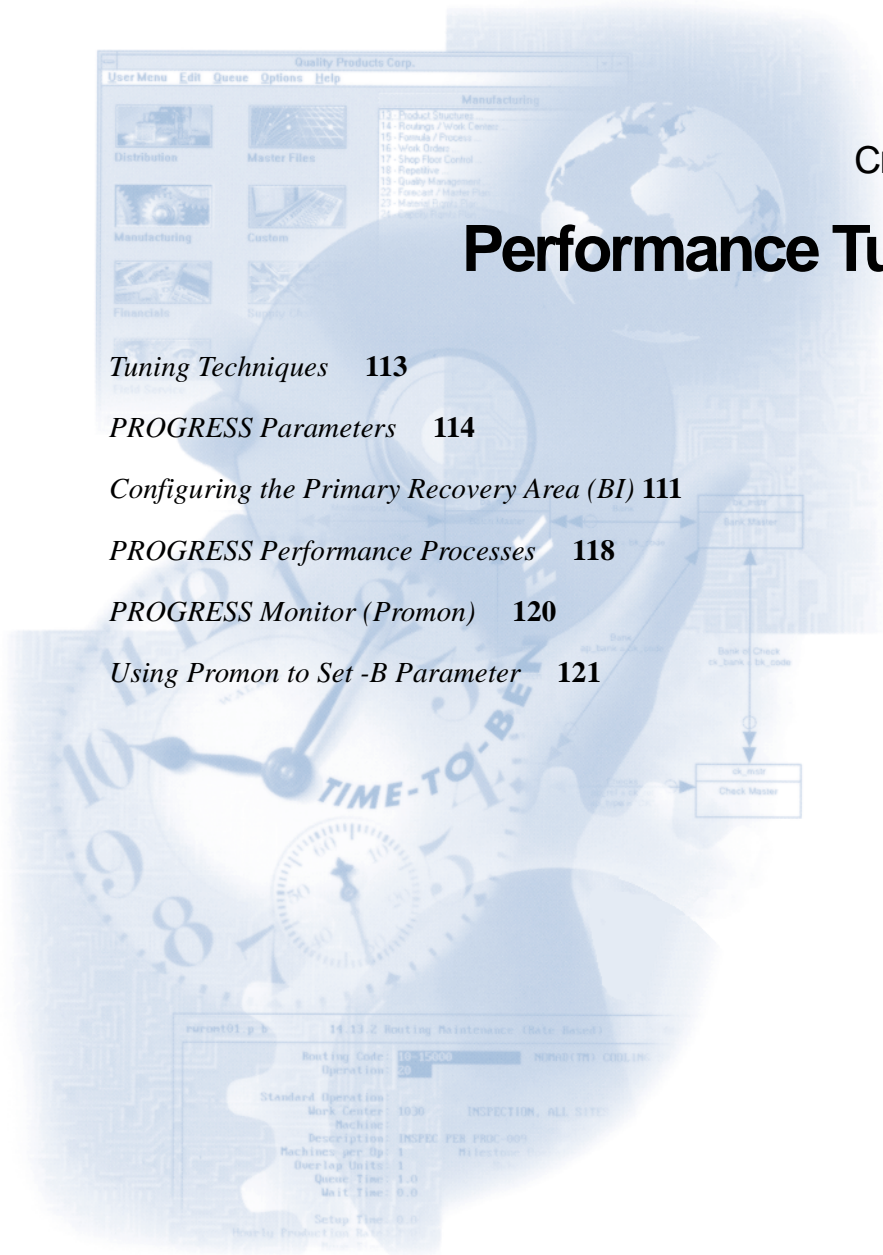
*PROGRESS Parameters* 114

*Configuring the Primary Recovery Area (BI)* 111

*PROGRESS Performance Processes* 118

*PROGRESS Monitor (Promon)* 120

*Using Promon to Set -B Parameter* 121



## Performance Tuning

- Progress Parameters
- Primary Recovery Area (BI) 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 (prowin32)	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 (_mprosrv)	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
-L	Lock Table	Server Start ( <code>_mprosrv</code> )	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 ( <code>_mprosrv</code> )	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.
-mmax	Maximum Memory	Client Start ( <code>prowin32</code> )	Specifies an allocation of memory for the end user. This memory is where compiled procedures can reside in memory, and scratch-pad work can be done.
-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.
-T	Temp Files	Client Start ( <code>prowin32</code> )	PROGRESS creates temp files, and sort files are created in the directory MFG/PRO was started from. You can redirect these files by using the -T followed by the directory.
-TB	Temp File Block Size	Client Start ( <code>prowin32</code> ) IDXBUILD	PROGRESS allocates a certain size to temporary files. When this size is exceeded, system operation halts and then reallocates a new size. By using -TB, 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.
-TM	Sort Speed	IDXBUILD	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 -by bypasses this menu for unattended shutdown operations.
-bibufs	BIW	Server Start ( <code>_mprosrv</code> )	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 Promon activity
-aibufs	AIW	Server Start ( <code>_mprosrv</code> )	Increases the number of after-image buffers in the after-image buffer pool. Default is 1. Initial value recommendation is 5, or 1.5 times the value of -bibufs
-Mf	BI	Server Start ( <code>_mprosrv</code> )	Delays PROGRESS from synchronously writing out to disk the last BI write at the end of a transaction. Default is 3 mS. Used to improve performance on heavily loaded shared memory systems. At system value, this value determines the oldest complete transaction that will be lost.

Option	Effect	Where To Set	Notes
-spin	Resource Waits	Server Start (_mprosv)	The number of times a process retries to acquire a latch on shared memory data structures. Default is 0. If initial value is greater than 0, a spin-lock algorithm is used which is more efficient than using Semaphores when the system has multiple processors. Initial value recommendation is 2000. Monitor CPU utilization and adjust accordingly. Resource waits can be viewed through Promon (R&D option)
-Ma	Remote Users	Server Start (_mprosv)	The maximum number of remote users per database server. Default is the -n/-Mn (# of users/Max Servers). This results in a value of 5 if defaults are used.
-Mn	Remote Users	Server Start (_mprosv)	The maximum number of database servers that will be spawned by the broker due to remote client access. Default is 4.
-Mi	Remote Users	Server Start (_mprosv)	Minimum number of clients on a database server before subsequent client access will spawn an additional database server. Default is 1.

## 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 *Primary Recovery Area* and *Creating a Multivolume Database* in the *PROGRESS System Administration Reference, Progress Version 9*.

## PROGRESS Performance Processes

- Before Image Writer (BIW)
  - Run one per database
  - Buffers a block of before-image writes, and commits the entire block at once
  - To start: `DLC\bin\probiw dbname` or use a Progress Utility
- After Image Writer (AIW)
  - Run one per database
  - Buffers a block of after-image writes, and commits the entire block at once
  - To start: `DLC\bin\proaiw dbname` or use a Progress Utility

### PROGRESS Performance Processes



See *Before Image Writers* and *Asynchronous Page Writers* 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` or use a Progress Utility
- Watchdog (Prowdog)
  - Run one per database
  - Identifies and eliminates “orphaned” processes on the database server after a set time
  - To start: `DLC\bin\prowdog dbname` or use a Progress Utility

## PROGRESS Monitor (Promon)

- Promon is a database monitoring utility
  - To start: `DLC\bin\promon.bat 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)



See *PROMON - The 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 in the database server start-up parameters
  - 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 *PROMON - The PROGRESS Monitor and Performance Tuning* 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

*Multi-Volume Overview*    **125**

*Multi-Volume Components*    **126**

*Multi-Volume Creation Steps*    **127**

*Creating a Structure File*    **128**

*Sample Structure File*    **129**

*Determining Database and Extent Size*    **130**

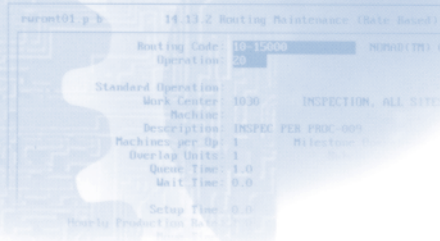
*Calculating the Structure File Extent Size Number*    **131**

*Using Prostruct*    **132**

*Creating a Void Multi-Volume Database*    **133**

*Populating a Void Multi-Volume Database*    **134**

*Copying a Populated Database to Void Database*    **135**



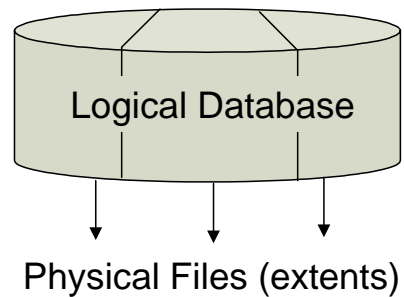
Routing Maintenance (Date Based)	
Routing Code:	10-15008
Operation:	20
Standard Operation:	1030
Work Center:	INSPECTION, ALL SITE
Machines:	1
Description:	INSPEC PER PROC 009
Machines per Sp:	1
Overlap Units:	1
Queue Time:	1.0
Wait Time:	0.0
Setup Time:	0.0

## Multi-Volume Database

- Components
- Creation
- Loading data

## Multi-Volume Overview

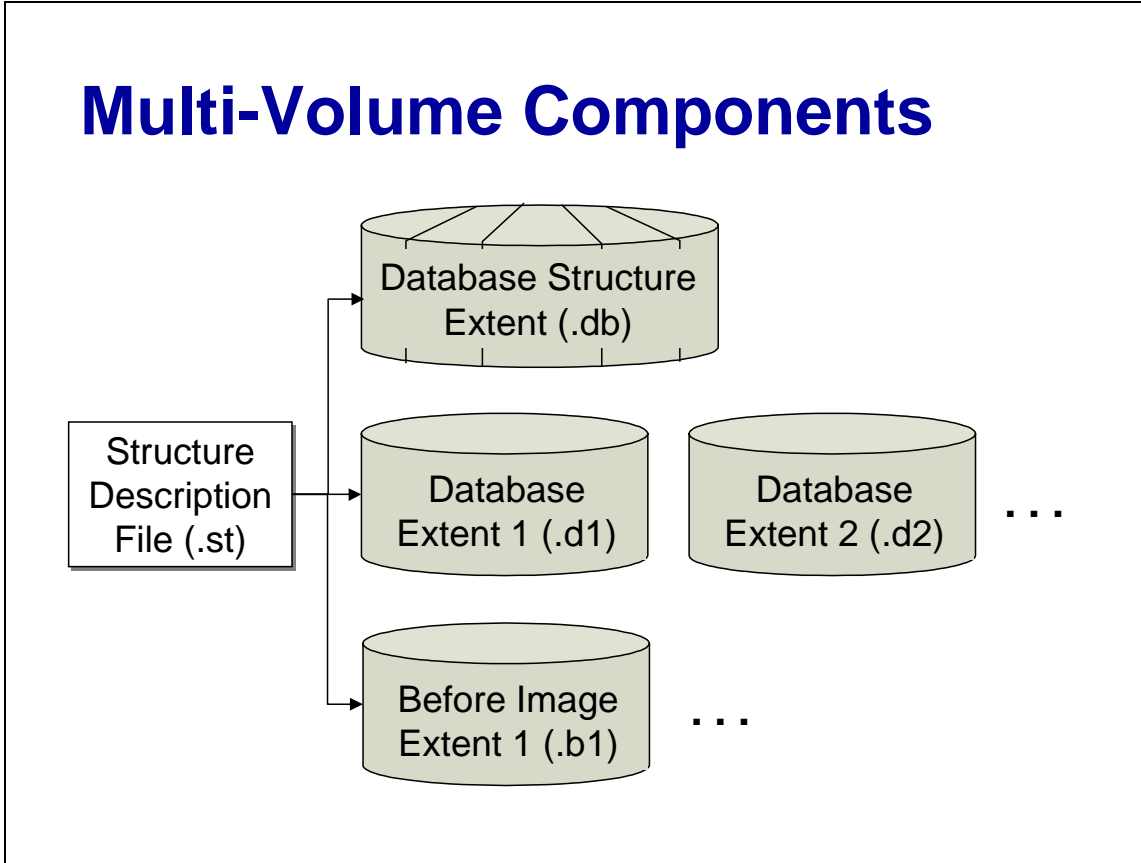
- One logical PROGRESS V9 database is comprised of several physical files
- These physical files can be placed across several physical disk drives to maximize I/O ability
- The before-image and after-image extents can also be made up of many physical files and located on specific I/O channels



### Multi-Volume Overview



See *Multi-Volume Databases* in the *PROGRESS System Administration Reference, Progress Version 9*



### Multi-Volume Components



See *Multi-Volume Components* in the *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 *Creating a Database* 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 a Structure 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 \usr\data1 f 5120	User Defined Area 7 dedicated to transaction data
d "TRANSACTION":7,64 \usr\data1 f 5120	User Defined Area 7 has two fixed extents
d "TRANSACTION":7,64 \usr\data1	**64 rec/blk, specific I/O channel, variable extent
#	
d "TRANSACTION_IDX":8,32 \usr\data2 f 5120	User Defined Area 8 holds transaction data indexes
d "TRANSACTION_IDX":8,32 \usr\data2	**Could have many extents & or many I/O channels
#	
d "STATIC":9,64 \usr\data3 f 5120	User Defined Area 9 holds static data
d "STATIC":9,64 \usr\data3	**Static data seldom changes = little I/O
#	
d "STATIC_IDX":10,32 \usr\data3 f 5120	User Defined Area 10 holds the static data indexes
d "STATIC_IDX":10,32 \usr\data3	**Grouped with the actual data since little I/O

### Sample Structure File



See *Creating a 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 1 – 2 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 *Multi-Volume Databases and Creating a Structure File* in the *PROGRESS System Administration Reference, Progress Version 9*



See *MFG/PRO Database Storage Areas* in the *MFG/PRO eB Installation Guide, PROGRESS for Windows NT*

## 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 *Multi-Volume Databases and Creating a Structure File* 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:

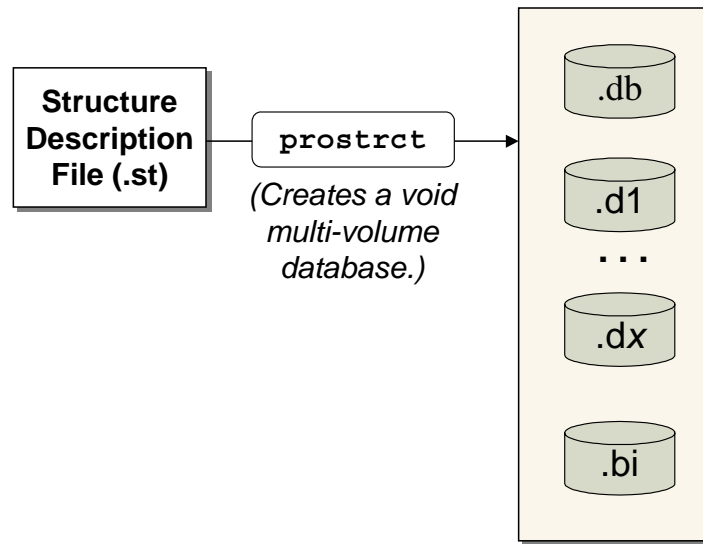
```
DLC\bin\prostrct create \dbdir\dbname \stdir\dbname.st
```

### Using Prostrct



See *Prostrct - The Multi-Volume Database Utility* 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.

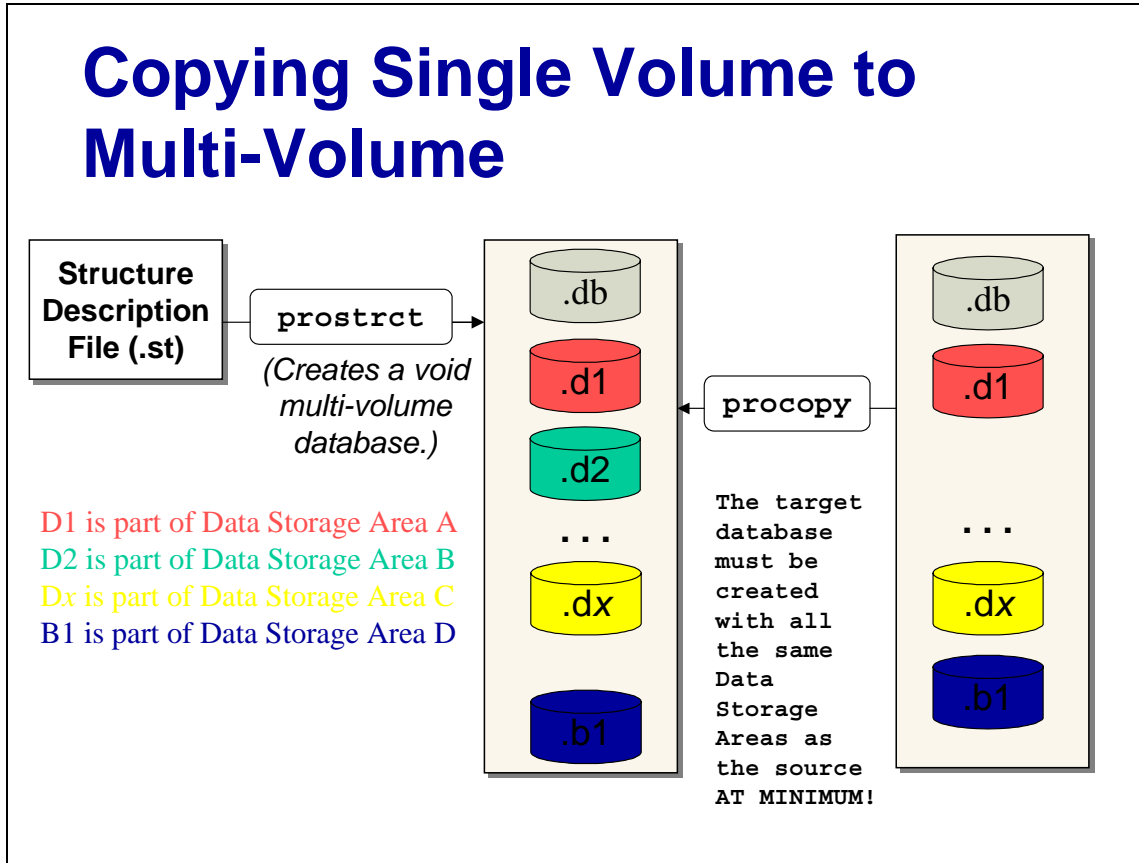
## Populating a Void Multi-Volume Database

- The newly created multi-volume structure is *void*
- Schema must be loaded by either method:
  - PROGRESS Administration 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 Administration, 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 *Procopy* in the *PROGRESS System Administration Reference, Progress Version 9*



### Copying a Populated Database to Void Database

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.

You can only use Procopy if the Target database has the same Data Storage Areas as the Source Database. The Target database may have other Data Storage Areas, but must contain all that were contained in the Source database at a minimum.



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



# System Utilities

*Database Set Administration*    **139**

*Compiling Programs*    **141**

*Batch Processing with the PROGRESS Batch Client*    **144**

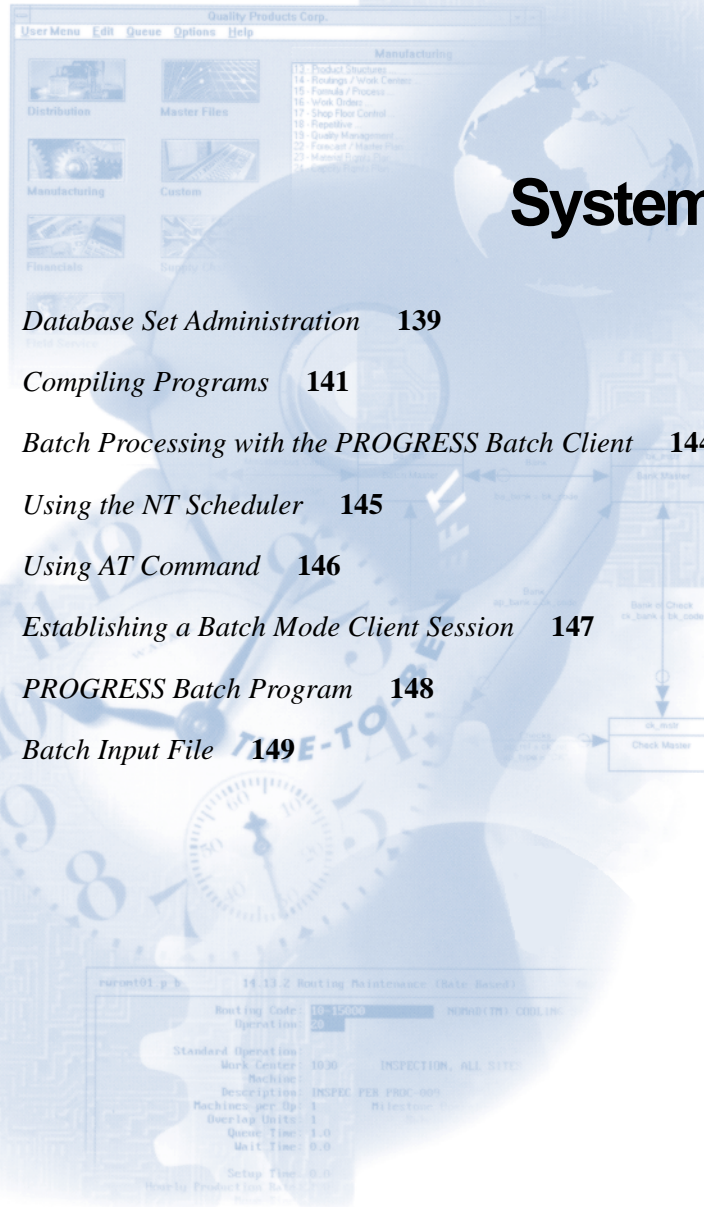
*Using the NT Scheduler*    **145**

*Using AT Command*    **146**

*Establishing a Batch Mode Client Session*    **147**

*PROGRESS Batch Program*    **148**

*Batch Input File*    **149**



## System Utilities

- MFG/UTIL
- AT jobs and the NT Scheduler
- 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
  - Client MFG/UTIL controls client session connections, and client code compilation
- There are separate client MFG/UTIL utilities to handle the character and Gui client connections
- Icons and parameter file contents are based on the database set definitions

### Database Set Administration



See *MFG/UTIL* and *Database Sets* earlier in this document



Complete the following Lab: “Reconfiguring Client Database Connections.”

## Database Set Administration (cont.)

- MFG/UTIL Procedure:
  - In MFG/UTIL, select the Configure menu and choose Any Database Set
  - Make changes in Database Set Configuration window
  - To incorporate changes, select the Scripts menu and choose one or both of the generate options
  - Make changes on both the server and Windows clients
  - If you added a new database name, add it to the hosts and services files on the server and client PCs

## Compiling Programs

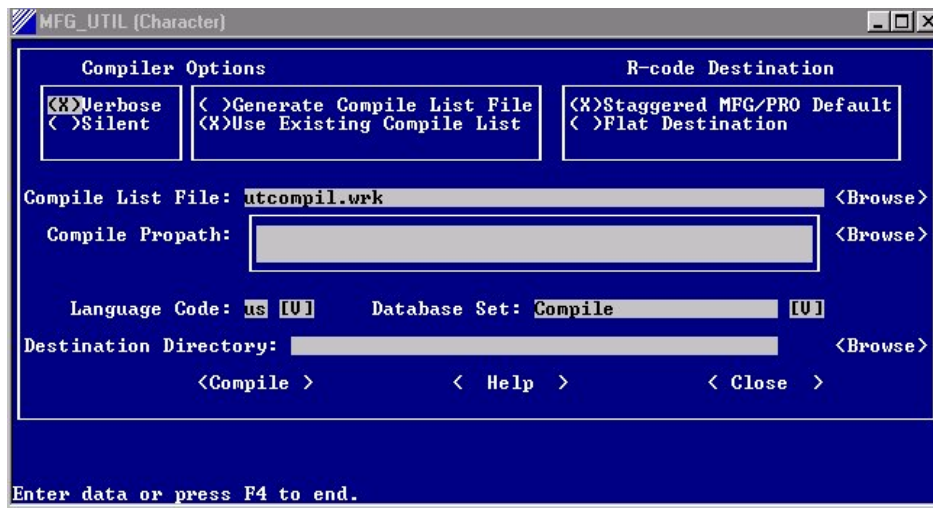
- Compile character and Windows code separately using separate MFG/UTILs
  - Specify source file directory or, if used, a work file
  - Specify destination directory for saved .r code if wanted
  - Use the compile database set to ensure that you compile with the qaddb logical database name
  - Set Propath 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

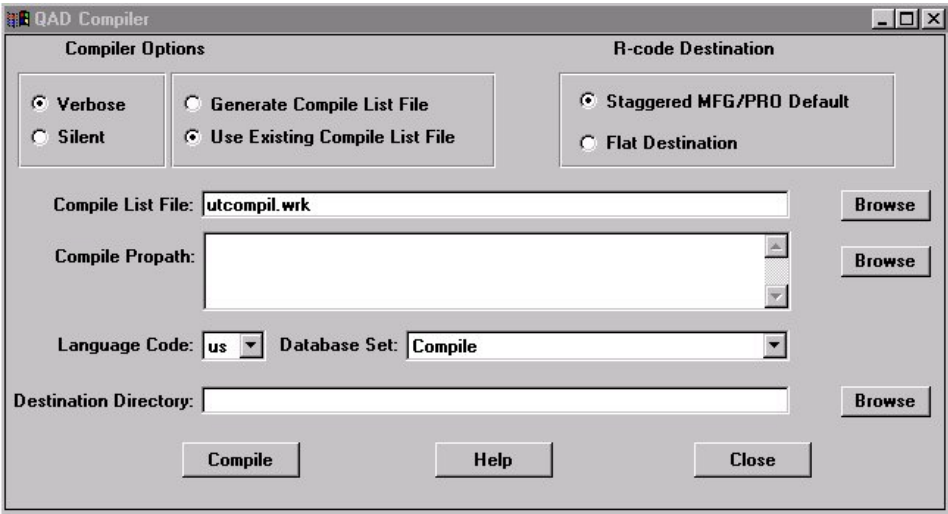


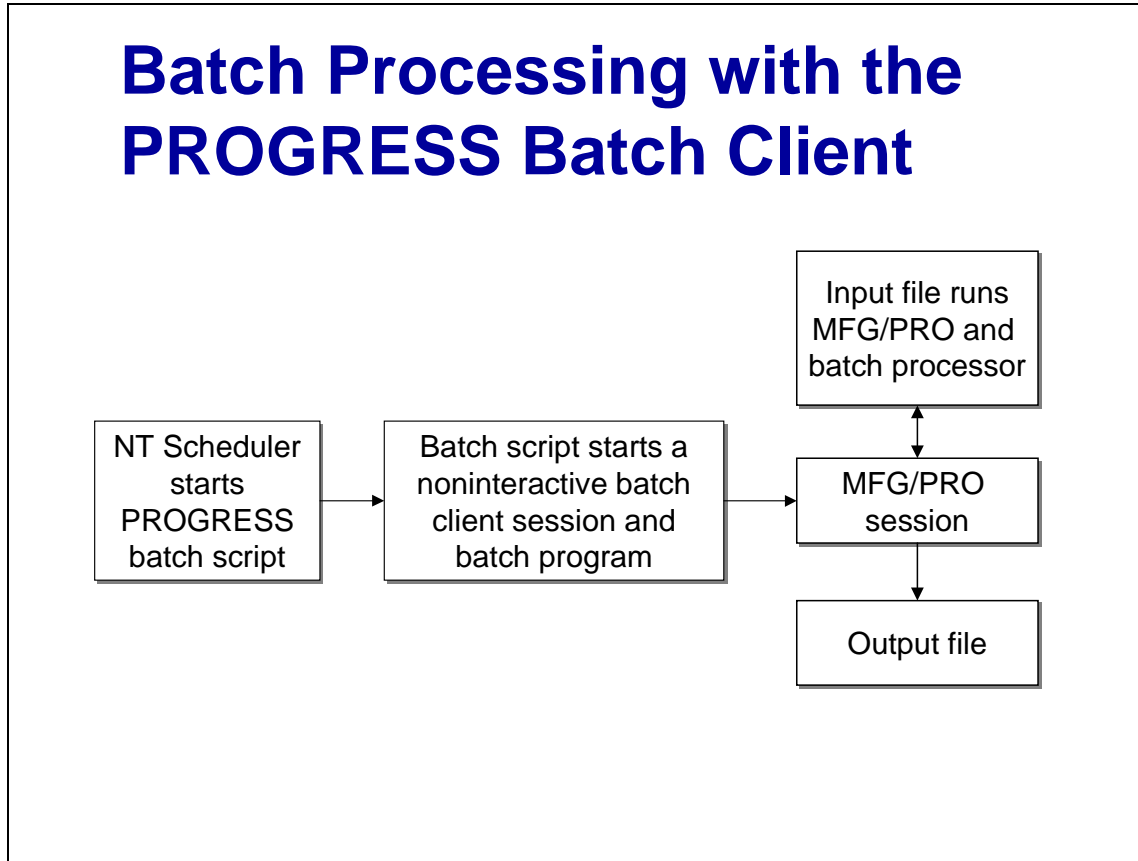
Complete the following Lab: “Compiling a Program.”

# Character MFG/UTIL Compile Screen



# GUI Client MFG/UTIL Compile Screen





### Batch Processing with the PROGRESS Batch Client

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

- Establish a noninteractive, 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 *Progress Batch Jobs* in the *PROGRESS Systems Administration Reference, PROGRESS Version 9*.

## Using NT Scheduler

- The Windows NT scheduler is a timer utility
- Used to issue commands at predetermined times
- Can be used to invoke a PROGRESS batch script
- Used with the AT command
- NT Administrator must have started the Control Panel->Services->Schedule service

### Using the NT Scheduler

For details on the NT Scheduler, refer to your Windows NT documentation.



Complete the following Lab: “Setting Up NT Scheduler Jobs for Batch Client Processing.”

## Using AT Command

- For a user who wants to automatically run a batch job every weekday, the syntax is:
  - MODEL: `at hh:mm /interactive /every:M,T,W,Th,F "command.bat"`
  - EXAMPLE: `c:>\ at 18:00 /interactive /every:M,T,W,Th,F "do_it.bat"`
- For a user who wants to automatically run a batch job later that day, and only once; the syntax is:
  - MODEL: `at hh:mm /interactive "command.bat"`
  - EXAMPLE: `c:\> at 18:00 /interactive "do_it.bat"`

### Using AT Command



For details on the *AT Jobs*, refer to your Windows NT documentation.



Complete the following Lab: "Using the AT Command."

## Establishing a Batch Mode Client Session

- Execute the DLC\bin\prowin32 executable
- Pass the -b and other applicable parameters
- Specify the database to connect to
- Specify a program to execute

```
dlc\bin\prowin32.exe \qad\dbname -db \qad\gui  
-ld qadgui -b -cdrcodein iso8859-1 -p batrun.p
```

### Establishing a Batch Mode Client Session



See *Progress Batch Jobs* in the *PROGRESS Systems Administration Reference, PROGRESS Version 9*.

## PROGRESS Batch Program

### BATRUN.P

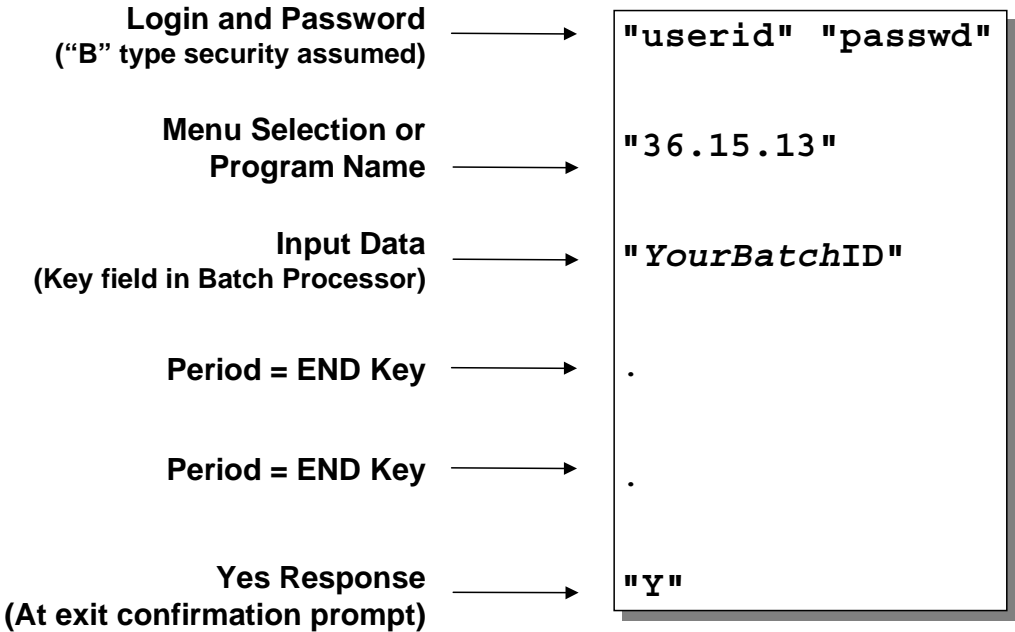
```
Propath = ". , \qad, \qad\us\src" .  
Input from \qad\bat.in.  
Output to \qad\bat.out.  
Run \qad\mf.p.  
Input close.  
Output close.
```

### PROGRESS Batch Program



See *Progress Batch Jobs* in the *PROGRESS Systems Administration Reference*, *PROGRESS Version 9*.

# Batch Input File



## Batch Input File



See *Progress Batch Jobs* in the *PROGRESS Systems 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

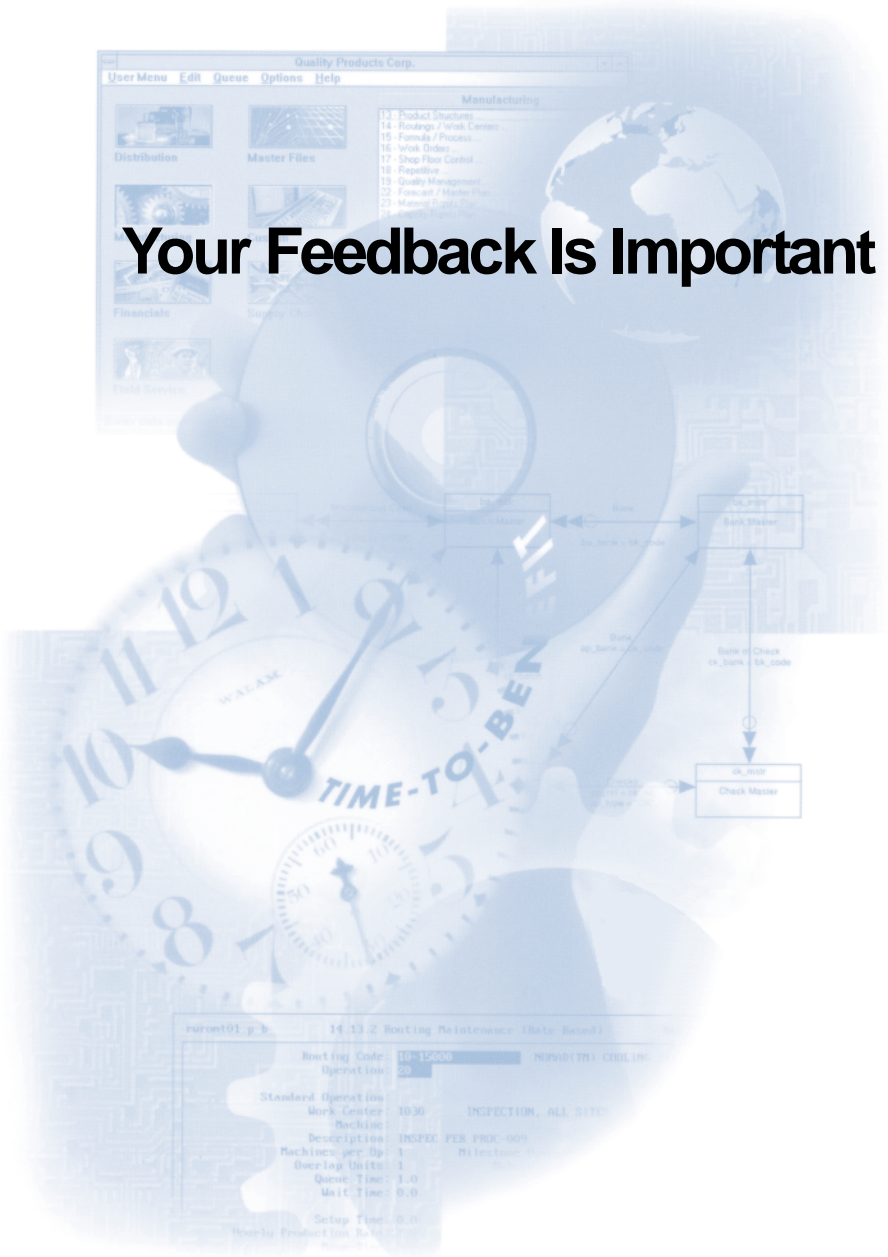


## **Sys Admin Progress on NT**

- ✓ Components and Configuration
- ✓ System Sizing and Installation
- ✓ Database Management
- ✓ Performance Tuning
- ✓ Multi-Volume Database
- ✓ System Utilities



**Your Feedback Is Important To Us**



parent01 y b 14.13.2 Routing Maintenance (Date Based)

Routing Code:	10-15000	NONP(TH) COIL-INS
Operation:	20	
Standard Operation:		
Work Center:	1030	INSPECTION, ALL SITS
Machines:	1	
Description:	INSPEC PER PROC-00%	
Machines per Op:	1	Reflection %
Overlap Units:	1	
Queue Time:	1.0	
Wait Time:	0.0	
Setup Time:	0.0	

Run by: Product Line 3, 12/15/00



*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](mailto:learningservices@qad.com)

Training Guide: eB System Administration Progress  
Database on Windows NT Server

Item number: 70-2822A

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](mailto:COPS@qad.com)

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



