

Industry-specific

QAD SOLUTIONS

Manufacturing Applications

MFG/PRO eB2.1 Installation Guide Enhanced Controls



78-0631A
MFG/PRO eB2.1 SP2 and Higher
January 2005

This document contains proprietary information that is protected by copyright and other intellectual property laws. No part of this document may be reproduced, translated, or modified 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.

QAD and MFG/PRO are registered trademarks of QAD Inc. The QAD logo is a trademark of QAD Inc.

Designations used by other companies to distinguish their products are often claimed as trademarks. In this document, the product names appear in initial capital or all capital letters. Contact the appropriate companies for more information regarding trademarks and registration.

Copyright ©2005 by QAD Inc.
78-0631A

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

Contents

| | |
|---|-----------|
| About This Guide | 1 |
| What Is In This Guide | 2 |
| Audience | 2 |
| Installation Documentation | 2 |
| Related MFG/PRO Installation Documentation | 2 |
| Related MFG/PRO eB2.1 Documentation | 3 |
| Document Conventions | 3 |
| UNIX and Windows Conventions | 3 |
| QAD Support Services | 4 |
| Support During an Installation | 4 |
| | |
| Chapter 1 Planning an Enhanced Controls Installation | 5 |
| Planning an Installation | 6 |
| Audit Databases | 7 |
| Administering Audit Databases | 7 |
| Audit Database Names and Directories | 9 |
| Enhanced Controls Installation Overview | 9 |
| | |
| Chapter 2 Enhanced Controls Install | 11 |
| Preliminary Steps | 12 |
| Add Enhanced Controls Services | 12 |
| Modify and Copy Audit Database Structure Files | 13 |
| Modify Workflow .ini Files | 15 |

- Installing Enhanced Controls 17
 - Populate OID Values 17
 - Enable Enhanced Controls 18
 - Creating Production Audit Databases 20
 - Loading E-Signature Data 22
- Creating an Enhanced Controls DB Set 23
- Generating Scripts and Shortcuts 26
 - Set the Correct PROPATH 26
 - Generate Enhanced Controls Scripts 27
- Compiling Application Code 29
 - Multithreaded Compiles 29
 - Compile MFG/PRO 30
 - Compile Additional Languages 34
- Starting Enhanced Controls from Batch 34
 - Create a Batch Client Startup Script 35
 - Create a CIM File 36
 - Create a Progress Batch Launch File 37
 - Running the Audit Process 37

Chapter 3 Audit Database Sizing..... 39

- Introduction 40
- Audit Database Default Values 41
- Progress Tools 41
 - Database Record Analysis 41
 - Storage Area Utilization 42
- MFG/PRO Database 42
 - Audit Trails 43
 - Electronic Signatures 44
- Storage Area Assignments 45
 - Audit Database 45
- Records per Block 46
 - General Recommendations 46
 - Audit Database 47

| | |
|-------------------------------|-----------|
| Extents | 48 |
| General Recommendations | 48 |
| Audit Database | 49 |
| Index | 53 |



About This Guide

What Is In This Guide 2

Installation Documentation 2

Document Conventions 3

QAD Support Services 4

What Is In This Guide

This guide covers how to:

- Install the Enhanced Controls module.
- Create audit trail databases.
- Administer and size audit trail databases.

Use this guide for all Enhanced Controls installations. For a service pack or conversion to eB2.1, complete the full eB2.1 install or conversion prior to running this installation.

Audience

These instructions are for a system administrator who is familiar with installing MFG/PRO, and is familiar with the UNIX operating system, the Microsoft Windows operating system, Progress software, and networking as necessary.

Installation Documentation

This document is available on the QAD Global Support Web site:

<http://support.qad.com/>

Before you start the installation, make sure you have the most recent version of this document.

The Enhanced Controls module is delivered on your MFG/PRO media. If you purchased this module, the necessary files are already in place once you have completed a standard MFG/PRO eB2.1 or eB2.1 Service Pack 2 installation. In addition, all standard installations of eB2.1 require that the empty audit database is created to support compiles as of Service Pack 2.

Related MFG/PRO Installation Documentation

The Enhanced Controls module requires a standard eB2.1 installation. For information on completing this install see one of the following:

- To install eB2.1, see *MFG/PRO eB2.1 Installation Guide: Progress Database*.

- To convert to eB2.1 from an earlier version of MFG/PRO such as eB2 or 9.0, see *MFG/PRO eB2.1 Conversion Guide: Progress Database*.
- To upgrade to the latest eB2.1 service pack from an earlier eB2.1 release, see *Installation Guide: MFG/PRO eB2.1 Service Pack X* where *X* is the latest service pack number.

Related MFG/PRO eB2.1 Documentation

The following books have been updated for the MFG/PRO eB2.1 release:

- *User Guide: MFG/PRO eB2.1 New Features* includes information about all changes included in the eB2.1 release, including Enhanced Controls.
- For technical details about MFG/PRO eB2.1, refer to the MFG/PRO eB2.1 *Entity Diagrams* and *Database Definitions*.

For a full list of other documentation, see the QAD Web site.

Document Conventions

This guide uses the conventions listed in the following table.

| If you see: | It means: |
|-----------------------------------|---|
| monospaced text | A command, file name, or path. |
| <i>italicized monospaced text</i> | A variable name for a value you enter as part of an operating system command; for example, <i>cdrom</i> . |
| indented command line | A long command that you enter as one line although it appears in the text as two lines. |
| Note | Exceptions or special conditions. |
| Important | Critical information. |
| Warning | Situations where you can overwrite or corrupt data, unless you follow the instructions. |

UNIX and Windows Conventions

This document supports the installation of Enhanced Controls on both UNIX and Windows platforms. The instructions use Windows screens and file and path conventions. In the few places where the two sets of instructions diverge, the headings and text state explicitly which operating system is the focus of the current set of instructions.

QAD Support Services

Enhanced Controls installations have a wide variety of configuration possibilities, are highly scalable, and are easily customized. While this guide provides basic installation information, it cannot consider every possible computing environment or configuration.

To take full advantage of the flexibility and potential of Enhanced Controls in your specific environment, contact your QAD Support representative for information on the installation and customization offerings supplied by QAD. These offerings include performance enhancements as well as technical and administration training.

Support During an Installation

There are resources in addition to this guide to help you if you run into problems.

Contact QAD Support Prior to Installation

Inform QAD Support that you are planning to install Enhanced Controls. Identify an individual who can be contacted if you run into problems. Discuss your hardware, disk space, and deployment plans with that individual.

Use the QAD Knowledgebase

On the QAD Web site, you can access QAD's extensive knowledgebase. Ask your questions in two or three ways to see if you get an answer closer to the problem you are experiencing.

Use Additional Web Resources

For Progress problems, use the Progress knowledgebase at their Web site:

http://www.progress.com/tech_support/knowledge_center/index.ssp

For OS-related problems, use standard vendor support, or use other search engines on the Web.



Chapter 1

Planning an Enhanced Controls Installation

This chapter presents basic topics that you should understand before beginning an Enhanced Controls installation.

Planning an Installation 6

Audit Databases 7

Enhanced Controls Installation Overview 9

Planning an Installation

Enhanced Controls provides important security and tracking functionality in MFG/PRO. The Enhanced Controls module lets you configure your system to maintain audit trails on key transactions and to use electronic signatures.

Audit trail records are created and stored in an MFG/PRO audit database. They contain facts about changes made in the MFG/PRO primary database. A typical audit record includes information that helps you identify who made a change, when the change was made, and what the change was. You can set up these functions for all MFG/PRO primary database tables or you can limit the audit trail recording activity to specific database tables. Currently only changes to tables in the `qadddb` database can be tracked.

The electronic signatures feature provides a means of attaching electronic signatures to automated processes. You can configure your system to require users of some MFG/PRO programs to enter a valid user ID and password before they can create or update records. Additionally, they must provide a reason code that defines the meaning of the signature; for example, Approved or Tested. Based on setup data, users may be able to enter a related remark as part of the signature.

These modules are part of an overall approach—including System Security and Enhanced Controls—to meeting the auditing requirements of MFG/PRO customers in regulated environments.

Enhanced Controls is an optional module and is licensed separately. Both the audit trails feature and electronic signatures require additional installation steps, and both use the separate audit database.

Audit Databases

Audit databases are required if you are implementing the optional Enhanced Controls module in MFG/PRO. To implement electronic signatures, one additional data load is required. The requirements for implementing audit trails are that you:

- Create a current audit database.
- Create a server script for the current auditing scenario.
- Create a plan and protocol for creating new audit databases.
- Create a plan and protocol for creating new server scripts in the future.

▶ For detailed information on implementing and using the audit functions, see “Enhanced Controls” in *User Guide: MFG/PRO eB2.1 New Features*.

Administering Audit Databases

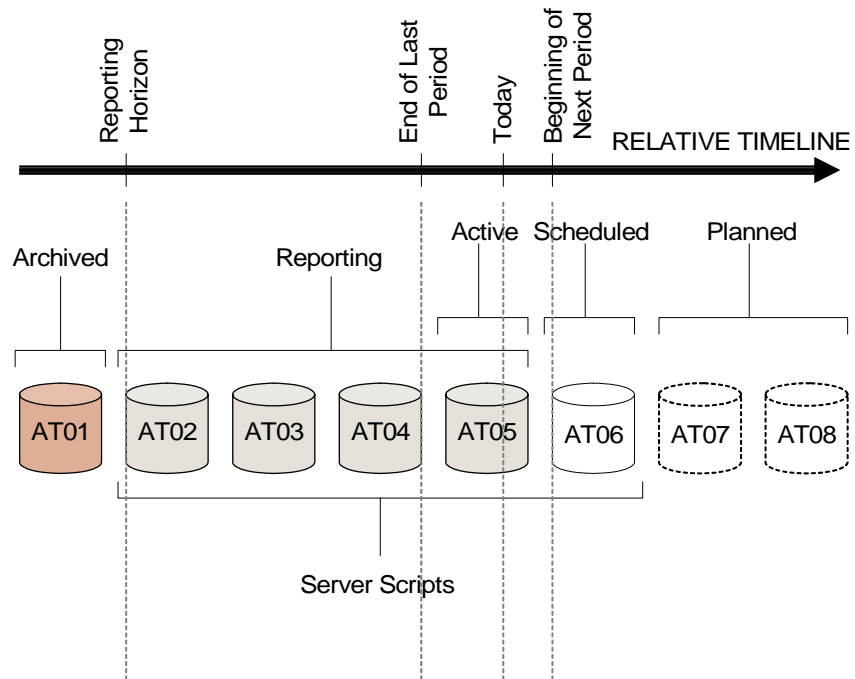
Audit databases are designed to fill up and eventually be taken offline. You set the date when an audit database is replaced using an MFG/PRO function. In turn, you also specify which new audit database comes online.

Audit information is always written to the current audit database. However, audit reports can be generated against past and current databases. Therefore, the administrator’s tasks include maintenance of past, current, and future databases, and the connections to these.

Using the example in Figure 1.1, assuming that several audit periods have passed, you would need servers running for all audit databases that may need a client connection. In the example, this would be databases AT02 through AT06. This includes all databases in the reporting horizon as well as the audit database that will take the place of the current one as required.

Database AT06 must exist and have a server running on the day the switchover is assigned in MFG/PRO.

Fig. 1.1
Audit Database
Example



Depending on how often your organization chooses to maintain the server scripts and databases, you may want to create several future audit databases, as well as several server scripts to address different periods. For example, the next server script in Figure 1.1 might launch AT03 through AT07.

Alternatively, you could create a server script for each audit database and then run multiple scripts when bringing up the system.

Audit databases are not specified for connection in the client scripts, as are the production, admin, and help databases. Instead, the connection is managed in MFG/PRO according to the setup data entered in Audit Database Maintenance (36.12.13.11). When audit information is written or when an audit report is run, the system automatically connects to the appropriate databases.

The audit databases can be included in your production database server scripts. This reduces the number of scripts to run, and the production database portion of the scripts can remain static. Alternately, you can create separate scripts that only launch the audit database servers. This is up to the administrators at each company.

Important You must have a new audit database available on the day you have assigned for the switchover in MFG/PRO. If a new audit database is not available, the audit information is written to a staging location in your production database, and this table can grow excessively.

Audit Database Names and Directories

QAD recommends either naming all audit databases identically and placing them in separate directories, or maintaining a simple naming scheme and optionally maintaining the databases in a single directory.

In either case, system administrators creating the audit databases and database sets, and generating startup scripts need to make sure the audit database workflows, database set definitions, and startup scripts have the correct path information.

Enhanced Controls Installation Overview

The process you will follow for the Enhanced Controls installation is included in MFG/UTIL workflows. During a standard MFG/PRO eB2.1 or eB2.1 Service Pack 2 or higher install, you use the Create Empty Progress Databases workflow, `wk0300.ini`, to:

- Create an empty audit database along with the other empty databases.
- Load schema into the empty databases.

You then create your production databases.

To install Enhanced Controls, you first enable Enhanced Controls in both the standard `mfgempty` database and in your main production database, `mfgprod`, using the Enable Enhanced Controls Schema Changes workflow, `wk0060.ini`. This workflow will:

- Validate OIDs in the target database.
- Load OID schema changes.

- Load audit trail schema changes.

Note OIDs are unique identifiers for each database record generated using a control value specified in Database Control (36.24).

You then run the Create Production Progress Audit Databases workflow, `wk0310.ini`, to:

- Create one or more new production audit databases.

You then load data for the electronic signatures tables. Following the data load, you integrate the production audit database with your MFG/PRO database sets to generate startup and shutdown scripts. You then recompile.

All of these steps are covered in this installation guide.

Enhanced Controls Install

Use this chapter to enable Enhanced Controls in your MFG/PRO empty and production databases, to create your production audit databases, to create startup and shutdown scripts for your Enhanced Controls environment, and to compile the MFG/PRO source code.

| | |
|--|-----------|
| <i>Preliminary Steps</i> | 12 |
| <i>Installing Enhanced Controls</i> | 17 |
| <i>Creating an Enhanced Controls DB Set</i> | 23 |
| <i>Generating Scripts and Shortcuts</i> | 26 |
| <i>Compiling Application Code</i> | 29 |
| <i>Starting Enhanced Controls from Batch</i> | 34 |

Preliminary Steps

Prior to starting an Enhanced Controls install, you must:

- Complete a standard MFG/PRO eB2.1 installation.
 - This loads the Enhanced Controls source code, schema, and data files onto the database server in preparation for this install.
 - It also creates the empty audit database.
- Add services for your audit databases.
- Modify and copy the production audit database structure file.
- Modify the Create Production Progress Audit Databases workflow.

Instructions for the last three steps are provided in this section.

Add Enhanced Controls Services

If you are implementing a client/server environment, tailor your `services` files. If you add any new service names after installation, you must modify your `services` file. All service names, host names, and port numbers must match in the `services` files of each client and each database server on the network. The location of the UNIX `services` file on the server is typically the `/etc` directory. On Windows it is located in:

```
c:\winnt\system32\drivers\etc
```

For Enhanced Controls, you will need a service for each audit database server you are running. For example, if you create monthly audit databases named `aud0105`, `aud0205`, `aud0305`, and so forth, and plan to connect to six month's worth of audit databases at a time—four past, one current, one future—you will need to define six services.

Add the names of your database services to your `services` file. Limit the service names to 16 characters. You can use any unused port numbers. Table 2.1 shows example service names and port numbers for a production database set and the added audit databases.

Table 2.1
Audit Database
Services Example

| Service Name | Port #/Protocol | Comment |
|--------------|-----------------|-----------------------|
| mfgprod-srv | 5500/tcp | # Production Database |
| admprod-srv | 5510/tcp | # Admin Database |
| hlpprod-srv | 5520/tcp | # Help Database |

| Service Name | Port #/Protocol | Comment |
|--------------|-----------------|--------------------|
| aud01-srv | 5530/tcp | # Audit Database 1 |
| aud02-srv | 5540/tcp | # Audit Database 2 |
| aud03-srv | 5550/tcp | # Audit Database 3 |
| aud04-srv | 5560/tcp | # Audit Database 4 |
| aud05-srv | 5570/tcp | # Audit Database 5 |
| aud06-srv | 5580/tcp | # Audit Database 6 |

Modify and Copy Audit Database Structure Files

QAD uses database structure files to define database names and the disk locations and sizes of database extents during database creation. You can edit the files during the database creation process, however, it is much easier to perform that task prior to launching the creation process.

Prior to launching the audit database creation workflow, you must:

- Edit the master `audprod.st` file for database extent and sizing.
- Create an `.st` file for each audit database you plan to create.
- Name each `.st` file exactly as the database name it will create.
- Edit each `.st` file for correct disk locations.

The standard `audprod.st` looks like:

```
#
b ./db
#
d "Schema Area":6,32 ./db
#
d "ATMSTR":7,64 ./db f 256000
d "ATMSTR":7,64 ./db
#
d "ATMSTR_IDX1":8,32 ./db f 128000
d "ATMSTR_IDX1":8,32 ./db
#
d "ATMSTR_IDX2":9,32 ./db f 128000
d "ATMSTR_IDX2":9,32 ./db
#
d "ATMSTR_IDX3":10,32 ./db f 128000
d "ATMSTR_IDX3":10,32 ./db
#
d "ATFDET":11,8 ./db f 2000000
d "ATFDET":11,8 ./db
#
#
d "ATFDET_IDX1":12,32 ./db f 128000
d "ATFDET_IDX1":12,32 ./db
```

```

#
d "ATFDET_IDX2":13,32 ./db f 128000
d "ATFDET_IDX2":13,32 ./db
#
d "ATKDET":14,64 ./db f 256000
d "ATKDET":14,64 ./db
#
d "ATKDET_IDX1":15,32 ./db f 128000
d "ATKDET_IDX1":15,32 ./db
#
d "ATKDET_IDX2":16,32 ./db f 128000
d "ATKDET_IDX2":16,32 ./db
#
d "ATKDET_IDX3":17,32 ./db f 128000
d "ATKDET_IDX3":17,32 ./db
#
d "AESIG":18,16 ./db f 256000
d "AESIG":18,16 ./db
#
d "AESIG_IDX":19,32 ./db f 256000
d "AESIG_IDX":19,32 ./db

```

As you can see, the file creates a default Schema Area and extents for four tables and nine indexes. The extents are paired: a fixed size extent, and a variable extent, and all extents are defined to be created in the *MFGPROInstallDir/db* directory.

A detailed discussion of audit database sizing is included in Chapter 3, “Audit Database Sizing,” on page 39. Use that chapter to estimate your audit database sizing requirements; then use those estimates to modify the base *audprod.st* file.

Creating a Structure File for Each Audit Database

Once you have adjusted the extent sizes in the base *audprod.st* file, you can copy that file, name the copies for the databases you are going to create, and modify the directory locations as necessary.

- 1 Copy *audprod.st* to *newDatabaseName.st*; for example, *aud0105.st*.
- 2 Open the new structure file and modify all instances of *./db* to reflect the desired target location of the new database extent. For example:

```

d "ATMSTR":7,64 ./db/aud0105 f 256000
d "ATMSTR":7,64 ./db/aud0105

```

Or:

```
d "ATMSTR":7,64 j:\mfgaudit\2005\aud0105 f 256000
d "ATMSTR":7,64 j:\mfgaudit\2005\aud0105
```

3 Save your changes.

All .st files should be saved in the directory where you intend to create the default Schema Area for the database. For example, if you are creating the database in j:/mfgaudit/2005/aud0105, place the edited structure file there as well.

4 Repeat steps 1 through 3 for all the production databases you intend to create.

Modify Workflow .ini Files

There are two Enhanced Controls workflows:

- One enables Enhanced Controls in the main MFG/PRO empty and production database schemas.
- One creates the new audit databases.

Only the create database workflow needs to be modified.

Modifying the Audit Database Create Workflow

By default, the Create Production Progress Audit Databases workflow, wk0310.ini, creates and truncates a single audit database named audprod.db. This workflow should be modified to create and truncate all the audit databases for which you have defined structure files.

The default workflow looks like:

```
WorkFlowDesc=Create Progress Production Audit DB

[CreateMSDB]
Status=Not Run
Program=crtpdbsa.w
STName=./db/audprod.st
FromDB=./db/audempty.db
;
;TRUNCATE PRODUCTION DATABASES
[TruncBI]
Status=Not Run
Program=truncbi.w
DBName=./db/audprod.db
Delay=0
;
```

A modified workflow to create three audit databases might look like:

```

WorkFlowDesc=Create New Audit Databases

[CreateMSDB]
Status=Not Run
Program=crtpdbsa.w
STName=j:/mfgaudit/2005/aud0105.st
FromDB=./db/audempty.db
;

[CreateMSDB]
Status=Not Run
Program=crtpdbsa.w
STName=j:/mfgaudit/2005/aud0205.st
FromDB=./db/audempty.db
;

[CreateMSDB]
Status=Not Run
Program=crtpdbsa.w
STName=j:/mfgaudit/2005/aud0305.st
FromDB=./db/audempty.db
;
;TRUNCATE PRODUCTION DATABASES
[TruncBI]
Status=Not Run
Program=truncbi.w
DBName=j:/mfgaudit/2005/aud0105.db
Delay=0
;
[TruncBI]
Status=Not Run
Program=truncbi.w
DBName=j:/mfgaudit/2005/aud0205.db
Delay=0
;
[TruncBI]
Status=Not Run
Program=truncbi.w
DBName=j:/mfgaudit/2005/aud0305.db
Delay=0
;

```

Note that the same empty database in the ./db directory is used each time. Save the modified workflow.

Warning If you change the workflow file name—to wk0311.ini for example—you must change the workflow title that appears on the first line of the file as well to avoid seeing a duplicate title in MFG/UTIL. This title is limited to 35 characters.

Installing Enhanced Controls

The following section provides instructions for enabling Enhanced Controls in your production databases, creating the audit databases, and loading e-signature data.

Populate OID Values

If you have completed a conversion or a service pack install rather than a new installation on Service Pack 2, you must populate OID values in the production database.

- 1 Launch MFG/UTIL on the database server.

```
./mfgutil
```

In Windows, double-click the GUI or character MFG/UTIL icon.

- 2 Select Populate OID Values from the Database|Enhanced Controls Options menu.
- 3 The Connect Database screen displays. Make sure you are connecting to your main production database, mfgprod. Choose OK.

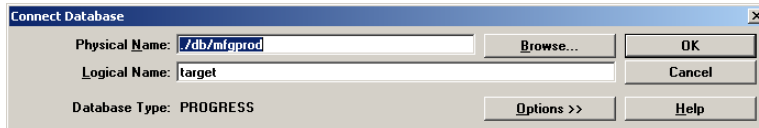
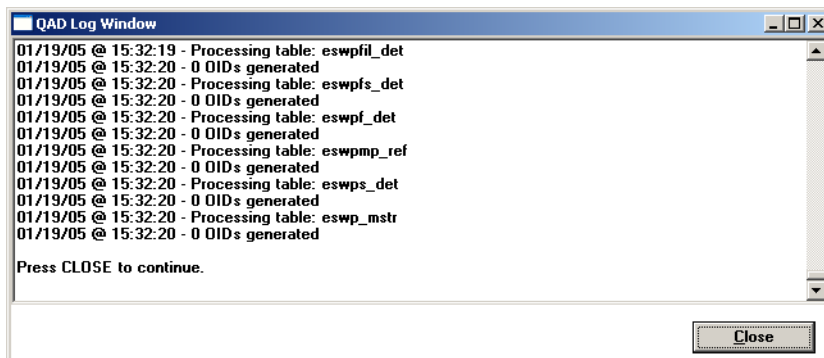


Fig. 2.1
Connecting to
mfgprod

- 4 The QAD Log Window displays while MFG/UTIL updates any uninitialized or missing OID values in your production database.

Fig. 2.2
OID Updates
Complete in Log
Window



- 5 Choose Close on completion.

Enable Enhanced Controls

This set of steps validates OIDs in the target database, then loads the two necessary audit trail schema files into the `mfgempty` and `mfgprod` databases.

- 1 Launch MFG/UTIL on the database server.

```
./mfgutil
```

In Windows, double-click the GUI or character MFG/UTIL icon.
- 2 Select MFG/PRO Guided Setup from the Configure menu. The Operation Sets screen displays.
- 3 In the Operation Sets screen, select Enable Enhanced Controls Schema Changes in the Operation Set drop-down list box.
- 4 Select Run Set and press Enter.
- 5 The Connect Database screen displays. Make sure this references `mfgempty.db`. Choose OK.

Fig. 2.3
Connect to
`mfgempty`



- 6 The QAD Log Window displays, showing the number of tables the program has checked for uninitialized OIDs and the number of tables that have them. There should be no tables with uninitialized OIDs. Choose Close to continue.

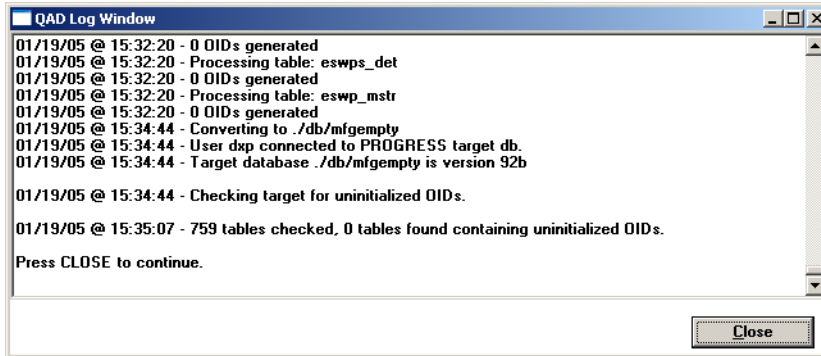


Fig. 2.4
No Uninitialized
OIDs in the Log
Window

- 7 The Connect Database screen displays again. Accept the defaults to connect to `mfgempty`. Choose OK.
- 8 The Load Data Definitions screen displays, showing the first data definition file, `mfg_optional_audit_trail_delta.df`.

Warning Choose Browse and manually select the correct file if you are installing after a service pack upgrade. In these instances, the default location is `MFGPROInstallDir\SPx\progrs\db` where `x` is the latest service pack number.

- 9 Choose OK to initiate the load.

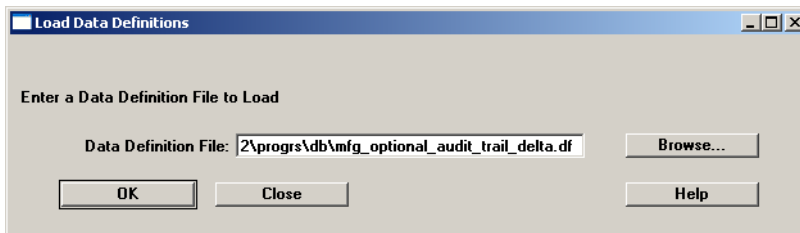


Fig. 2.5
Load Data
Definitions

- 10 The Log Window opens to show progress. Wait until the log displays the message Schema Load Complete, then choose Close.

- 11 The Connect Database screen displays again. Accept the defaults to connect to `mfgempty`. Choose OK.
- 12 The Load Data Definitions screen displays again, showing the second data definition file, `./db/mfg_optional_oid_delta.df`. Choose OK to initiate the load.

Warning Choose Browse and manually select the correct file if you are installing after a service pack upgrade. In these instances, the default location is `MFGPROInstallDir\SPx\progrs\db` where *x* is the latest service pack number.

- 13 The Log Window opens. Wait until the load is complete; then choose Close to continue.
- 14 The Connect Database screen displays. Make sure you connect to the next database to be updated. Choose OK.

Warning If you have not already edited workflow `wk0060.ini` to add `mfgprod`, you can do that now, changing the instances of `mfgempty` to `mfgprod`, and setting all Status values to Not Run.

- 15 Repeat steps 6 through 13 for the `mfgprod` database.

Creating Production Audit Databases

In this task, use the empty audit database, `audempty`, and the structure files you created in “Creating a Structure File for Each Audit Database” on page 14, to create production audit databases. Cycle through the entire process for each new audit database defined in the modified workflow from “Modifying the Audit Database Create Workflow” on page 15.

Warning If you are installing after a service pack upgrade, use the Browse button in each dialog to verify the database and file paths. The default location for structure files is `MFGPROInstallDir\SPx\progrs\db` where *x* is the latest service pack number. For databases, the default path is `MFGPROInstallDir\db`.

- 1 Choose Guided Setup from the Configure menu.
- 2 Select Create Progress Production Audit DB in the Operation Set drop-down list. Choose Run Set.

- 3 The QAD Database Builder screen displays. A structure file for the first audit database, for example `aud0105.st`, displays in the Structure File field. The MFG/UTIL screen that displays lets you assign disk locations and sizes to your storage areas.

Important For production audit databases, QAD strongly recommends that you make all changes to storage areas in the structure file itself as described in “Creating a Structure File for Each Audit Database” on page 14.

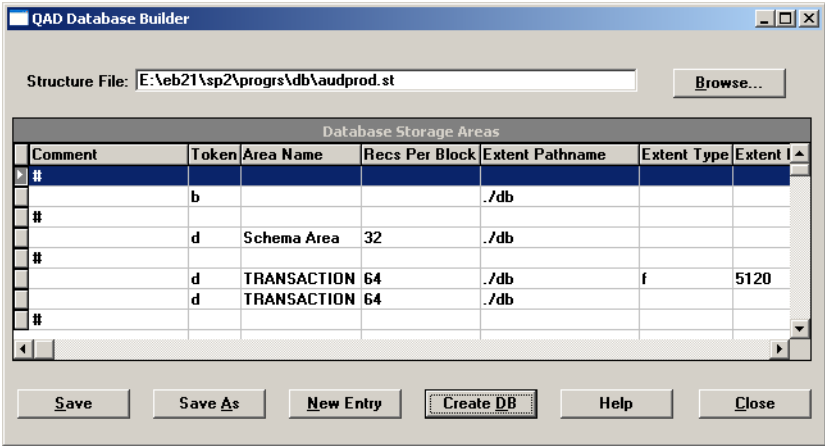
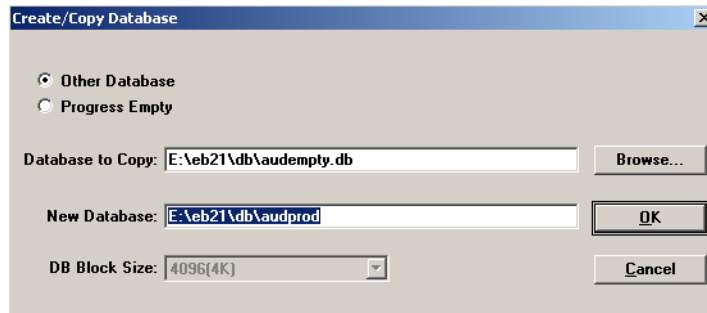


Fig. 2.6
QAD Database
Builder for
mfgprod

- 4 Choose Create DB to continue.
- 5 The Create/Copy Database screen displays. Accept the default of Other Database and verify the path to `audempty.db`. The New Database name defaults from the `.st` file name identified in the workflow. Choose OK.

Fig. 2.7
Create/Copy
Database screen for
audprod



- 6 When the audit database is built, a log of the build process displays. Scroll through the log and verify that the database was created successfully. This information is also recorded in *DBName.log* in the *MFGPROInstallDir*. When ready, choose Close to continue.
- 7 The Database Storage Areas screen displays with the Close button selected. Press Enter or choose Close to close the screen.
- 8 The Truncate BI File screen displays. Accept the default path to the audit database and choose Truncate.
- 9 Close the Log Window that displays on completion.
- 10 Steps 3 through 9 repeat for each audit database you have defined in the modified workflow.

Loading E-Signature Data

Several data files must be loaded for the electronic signature feature.

- 1 In MFG/UTIL, select Database|Enhanced Controls Options|Load Enhanced Controls Data. The Database Connect screen displays.
- 2 Make sure to connect to your main production database, *mfgprod*. Choose OK.
- 3 The QAD Log Window displays to show the connection. Choose Close.

- You are prompted for the input directory. Browse to locate the *MFGPROInstallDir/mfg* directory. Following a service pack install, this directory will be *MFGPROInstallDir/SPx/mfg* where *x* is the latest service pack number. Select any file in the directory and choose Open.

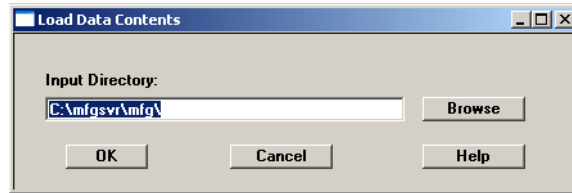


Fig. 2.8
Input Directory for
E-Signature Data
Load

- Choose OK to initiate the load. The QAD Log Window displays on load completion.

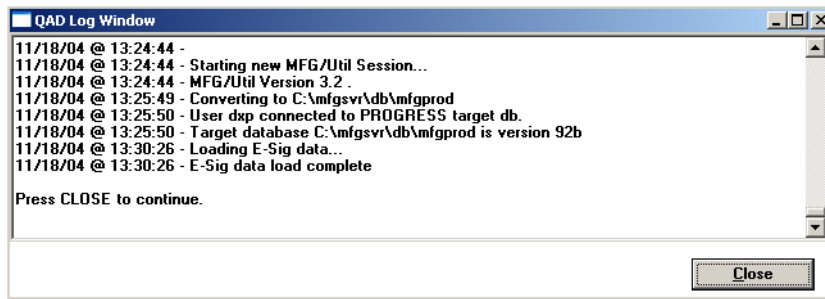


Fig. 2.9
E-Signature Data
Load Confirmation

- Choose Close to continue.

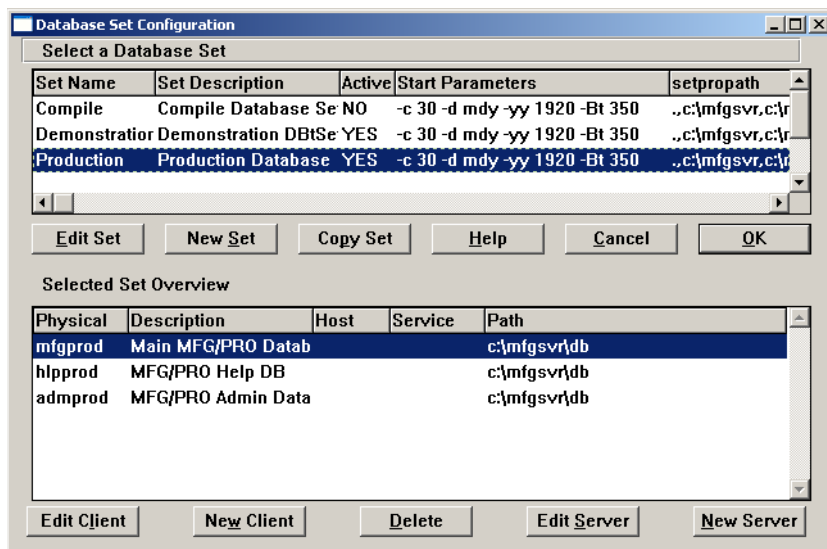
Creating an Enhanced Controls DB Set

During a standard MFG/PRO eB2.1 install, you should have two database sets defined, at a minimum. The first is the Production database set; the second is a Compile set. The Compile database set should already include the empty audit database because it is required for all eB2.1 compiles. The following steps cover how to copy the Production database set and add as many of the new audit databases as you need to get started.

Important If you are using Enhanced Controls in a demonstration or training environment, create separate audit databases for these environments to ensure the purity of the production audit databases.

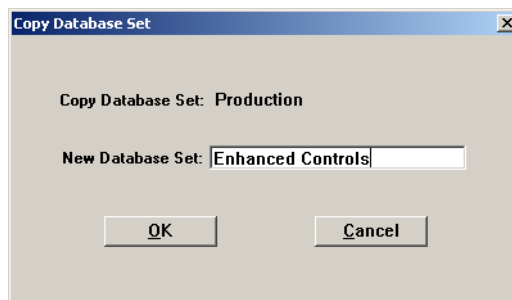
- 1 In MFG/UTIL, select Configure|Database Set Maintenance. The Database Set editor opens.
- 2 Select the Production database set.

Fig. 2.10
Production
Database Set



- 3 Choose Copy Set and enter the new database set name.

Fig. 2.11
Copying the
Production
Database Set



- 4 Make sure the Enhanced Controls database set is selected, then choose New Client to add a new database.

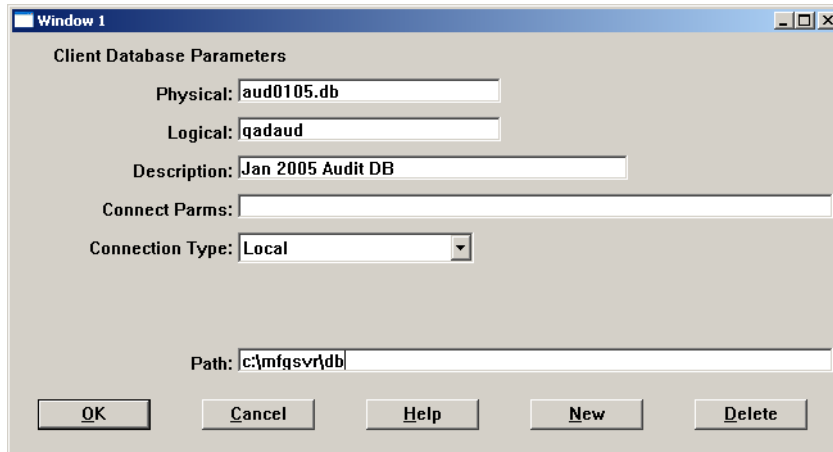


Fig. 2.12
Adding a New
Client Database

- 5 Enter the physical name, logical name, and the client/server attributes of the database. The logical name must be qadaud for all audit databases. Choose OK to save the information.
- 6 Choose Edit Server to update the server parameters.

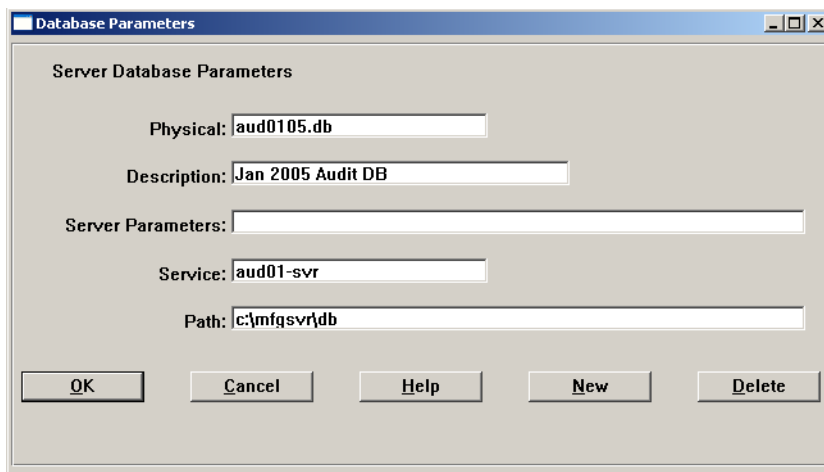


Fig. 2.13
Audit Database
Server Parameters

- 7 Enter a valid service name, as entered in the `services` file in “Add Enhanced Controls Services” on page 12.

- 8 Choose OK to continue.
- 9 Repeat steps 4 through 8 for each audit database you want to add to the Enhanced Controls database set.
- 10 When you have added all the audit databases, choose OK in the Database Set Maintenance screen to save your changes.

Generating Scripts and Shortcuts

The next steps create startup and shutdown scripts or shortcuts for the new database set. You can edit these scripts and icons after you generate them if you choose. There are three different types of scripts or icons:

- UNIX scripts
- Windows character session icons
- Windows GUI session icons

The following instructions cover all three types.

Set the Correct PROPATH

When configuring database sets using MFG/UTIL, a value defaults to the PROPATH field. Depending on the specific system configuration, the default value may not be correct. Continuing an MFG/UTIL operation without updating the PROPATH can cause errors. Typically, the PROPATH value defaults from the value defined in the Paths for MFG/PRO and MFG/UTIL screen.

Note the following points when working with MFG/UTIL:

- In MFG/UTIL, select Configure|Set Paths for MFG/PRO and MFG/UTIL. Review and update the values as needed.
- When using MFG/UTIL to generate any script or complete any task, always carefully review the PROPATH values displayed for each screen you access.
- When working with database sets, verify the PROPATH for each database in the Database Set Parameters screen. Access this screen by choosing Edit in the Select a Database Set screen.

- The typical PROPATH for a new MFG/PRO installation includes these directories in this order:
 - a The current directory (.)
 - b (For service pack installs only) The latest service pack install directory (c:\mfgsvr\SPx)
 - c The directory where the client code was installed (c:\mfgsvr)
 - d The xrc directory

Example .,c:\mfgsvr\sp2,c:\mfgsvr,c:\mfgsvr\xrc

Generate Enhanced Controls Scripts

- 1 Select Scripts|Generate Scripts in MFG/UTIL. The Server Script Creation screen displays.

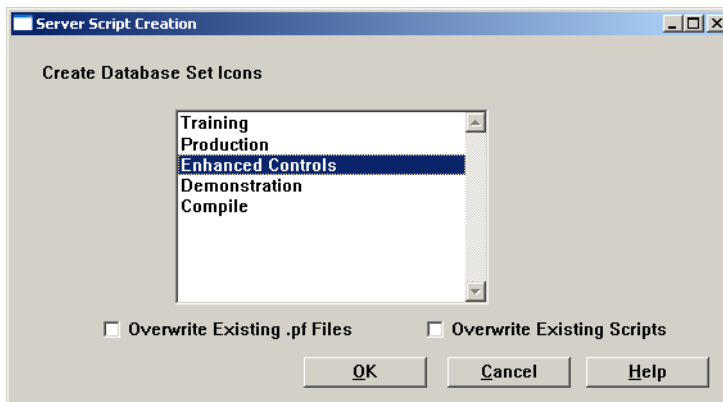


Fig. 2.14
Server Script
Creation Screen

- 2 Press spacebar or click in Windows to select the new Enhanced Controls database set and choose OK.
- 3 For Windows shortcuts, select the program folder where you want the shortcuts stored.
- 4 In Windows, you view a confirmation screen. Click Next to generate the shortcuts.

Note The Windows icon build is completed by the `makeicon.exe` program. This program must be in your `PROPATH`. It is located in `MFGPROInstallDir` by default.

- 5 For UNIX systems, MFG/UTIL creates the scripts listed in Table 2.2 for each database set. Use these scripts, or copies of them, for all startup and shutdown processes. The scripts are created in the `MFGPROInstallDir`.

Table 2.2
UNIX Scripts
Generated by
MFG/UTIL

| Script | Description |
|-------------------------------|---|
| <code>start.DBSetName</code> | This script starts servers for the databases in the set. <code>start.Enhanced Controls</code> starts servers for <code>mfgprod.db</code> , <code>admprod.db</code> , <code>hlpprod.db</code> , and any audit databases you added to the set. |
| <code>stop.DBSetName</code> | This script shuts down servers for all databases in the set. |
| <code>client.DBSetName</code> | This script starts an MFG/PRO character client session in multiuser mode. This script functions only after you install the MFG/PRO character clients. This can be deleted for the Enhanced Controls database set: audit databases are always connected to through an active MFG/PRO client. |
| <code>qma.DBSetName</code> | This script is created only for UNIX servers and is used by MFG/PRO QAD Desktop clients. This script is also unused for audit databases and may be deleted. |
| <code>qma2.DBSetName</code> | This script is created only for Tomcat servers and is used by MFG/PRO QAD Desktop clients. This script is also unused for audit databases and may be deleted. |

Windows shortcuts are Start and Shutdown for each database set.

- 6 After the scripts or shortcuts are generated, choose Close in the Installation Log Window.
- 7 Review the log file, `mfgutil.log`, for any errors.

Compiling Application Code

A full compile of the server code is required after installation of Enhanced Controls. These instructions replace the compile steps in the standard MFG/PRO eB2.1 installation.

Compiling a source file creates an object file with the same name and the `.r` extension. Compiled programs are saved into a subdirectory using the first two letters of the program name. This subdirectory is located below the two-letter language code directory below *MFGPROInstallDir*. For example, `sosomt.p` compiled for U.S. English is placed in *MFGPROInstallDir/us/so*.

Important If you are compiling for a non-English language, you must first create a language-specific compile database set as described in your MFG/PRO eB2.1 installation guide.

Multithreaded Compiles

Compiles take several hours. To substantially reduce compile time, you can set up the compile to multithread using multiple compile listings and by initiating multiple compile sessions. The following steps summarize details in the next section:

- 1 In the Generate Compile Listing screen, use the From and To fields to limit the scope of the file you are generating. Enter a unique file name.
- 2 Run Generate Compile Listing again for the next set of files.
For example, assuming you compile every program (7746 files), you could include programs from A to FO in `utcomp01.wrk` (1736 files), FS to H in `utcomp02.wrk` (1882 files), I to RC in `utcomp03.wrk` (2007 files), and RE to Z in `utcomp04.wrk` (2120 files).
- 3 You then launch separate, concurrent MFG/UTIL sessions for each compile process. Launch them all concurrently. Four sessions save from 50% to 75% of the processing time.

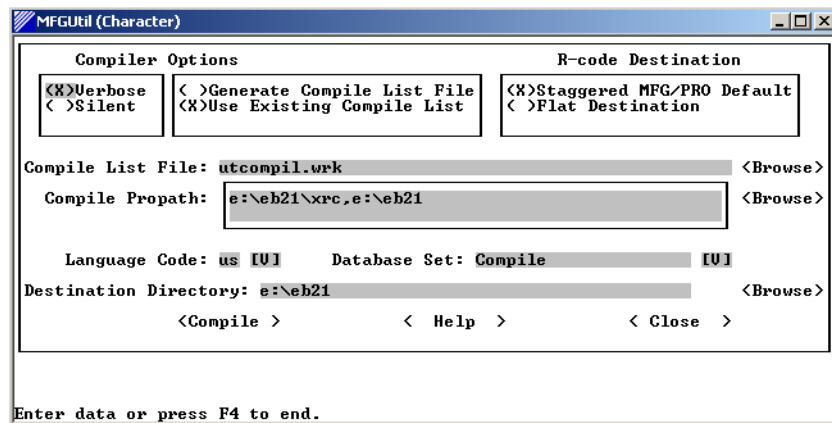
Important Errors display in each compile session stating that the process cannot open `mfgutil.log`. This occurs when another process is writing to the file. The file is eventually updated and the errors do not harm the compiles.

Compile MFG/PRO

The files MFG/UTIL compiles are those listed in the file `utcompil.wrk`. By default, this file lists the files shipped with MFG/PRO. You can change the file listing in `utcompil.wrk` during the process of setting up for the compile, but this is usually done only for subsequent compiles for custom development.

- 1 Using the Guided Setup operation set, the Compiler Options screen displays automatically. Otherwise, choose Compile Procedures from the Programs menu.
- 2 Use the screen example and field descriptions to select compile options. The Compiler Options in the upper left of the screen let you set feedback levels and select a compile list.

Fig. 2.15
QAD Compiler



Verbose. MFG/UTIL displays compile information on the screen and writes it to the MFG/UTIL log file (`mfgutil.log`). During the compile, the following information displays:

- Date and time
- Percentage of the compile completed

- Path and name of the program currently compiling
- Number of compile errors that occur

The log file is in the directory from which MFG/UTIL was launched.

Silent. MFG/UTIL writes to `mfgutil.log` only.

Generate Compile List File. Generates or regenerates a file listing the programs to compile. When you select this option, the Generate Compile List screen displays.

Use Existing Compile List File. By default, this is `utcompil.wrk`, which is shipped with MFG/PRO and contains the full list of files.

Staggered MFG/PRO Default. Saves compiled code in the default structure of language directories underneath `MFGPROInstallDir`.

Flat Destination. Select this option to save the compiled code in a single destination directory.

Compile List File. Specify the name of the compile list file, by default `utcompil.wrk`. If the file is located in a directory other than the one from which MFG/UTIL was launched, include the directory path and the file name.

Compile Propath. The compile PROPATH must contain:

- The `MFGPROInstallDir`
- The `xrc` subdirectory in `MFGPROInstallDir`
- The language-specific directory containing the source code for the programs you are compiling
- The language-specific directory containing any include files for the source code you are compiling

Language Code. If you selected Staggered MFG/PRO Default option, enter the language code where you want the code saved.

Database Set. The database set against which to compile.

Destination Directory. Specify the directory where you want compiled code saved. This is typically the `MFGPROInstallDir`. If Staggered MFG/PRO Default is selected, the compile places compiled code in

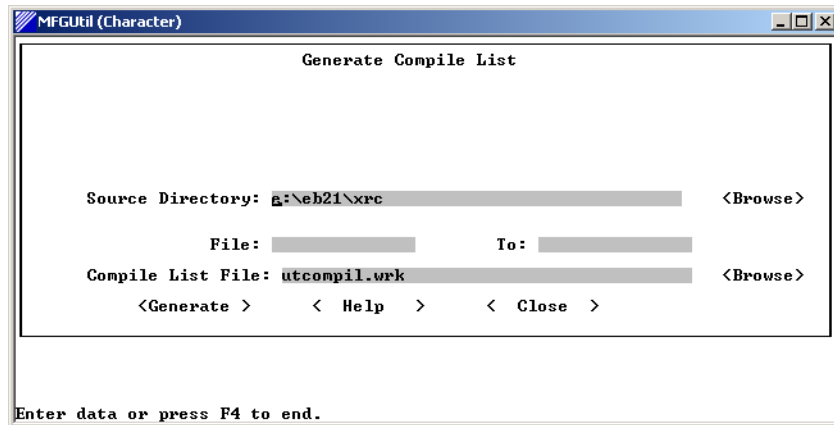
▶ See “Compiling Application Code” on page 29.

the appropriate language directory beneath this directory and within two-letter directories beneath that. For example, a U.S. English `sosomt.p` is compiled to:

```
./MFGPROInstallDir/us/so/sosomt.r
```

- 3 If you selected the Generate Compile List File option, complete the following instructions. If you selected Use Existing Compile List File, skip to step 5.

Fig. 2.16
Compile List
Generator



a In the Source Directory field, specify the path to the directory containing the source programs you want to compile. By default, this is `MFGPROInstallDir/xrc`.

b To compile a subset of the programs, specify the subset using the File and To fields (limited to alphanumeric sorts).

c Specify the file name to use in the Compile List File field. The default file name is `utcompil.wrk`.

If you want the list file created in a specific directory, include the path. If you do not specify a path, the file is generated in the directory from which MFG/UTIL was launched (usually `MFGPROInstallDir`).

d Choose Generate to create the compile list file.

e The Compiler Options screen displays with the Use Existing Compile List option selected. Enter the correct compile list name.

▶ See
“Multithreaded
Compiles” on
page 29.

- 4 When ready, choose Compile.
- 5 In the compile verification screen, verify the compile information. If the compile settings are correct, choose Continue. If the settings are incorrect, choose Back to make changes.

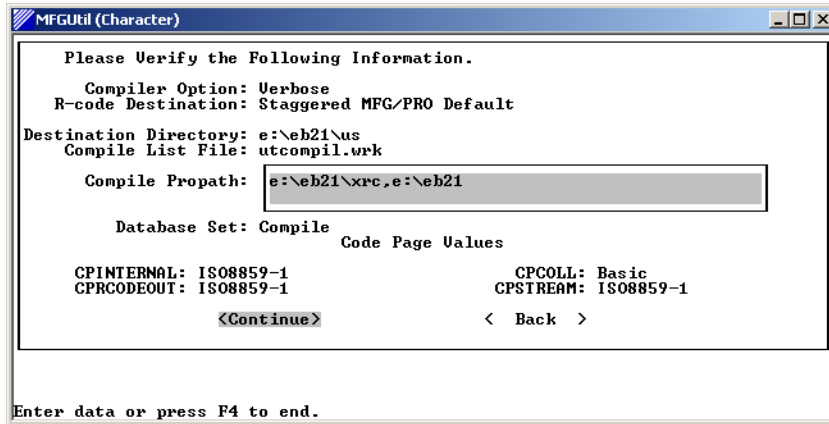


Fig. 2.17
Compiler Summary

Note If you are compiling code for a non-Western European language, you may have to modify the code page values for the compile. The compile code page values default from the `startup.pf` file located in the Progress directory.

- 6 If you selected Verbose, an Installation Log screen displays. When the compile ends, choose Print or Close.
- 7 Once the compile completes, check the MFG/UTIL log file for errors. In the event of errors, correct the errors and rerun the compile.

If you encounter compile errors, the most prevalent errors are:

- Incorrect PROPATH is used.
- PROPATH is missing the `/src` or `/xrc` directories.
- Progress database version does not match the code version.

Note MFG/UTIL saves the settings entered in the QAD Compiler screen so that you can reuse or modify them for the next compile.

Compile Additional Languages

If you are implementing additional languages, you can now load the translated labels for the first of these languages and recompile. The sequence to follow is:

- 1 Compile in the base language.
- 2 Load translated labels for next language.
- 3 Compile code for next language.
- 4 Repeat steps 2 and 3 for additional languages.

Starting Enhanced Controls from Batch

In order to automate the startup of Enhanced Controls and begin extracting audit data from your production databases, you must copy and modify a production client startup script, then launch all the components in the correct order.

Depending on the number of users and the number of tables being audited, you may want multiple background processes running. Each background process consists of a client session that is running Audit Trail Creation Process (`attpui.p` or 36.12.13.23).

To start these processes, you create a startup script that launches a batch program, which in turn launches a CIM file. The CIM file automatically starts the Audit Trail Creation Process program.

The advantage of this method is that you do not have to start Audit Trail Creation Process sessions manually. However, the CIM file contains an unencrypted user name and password. If you choose to follow this method, make sure the user is defined in MFG/PRO with access only to the single Audit Trail Creation Process menu item.

To use a batch startup process, follow the steps outlined below. Details are provided in subsequent sections.

Implementation

- 1 Create a batch client startup script.
- 2 Create a CIM file.
- 3 Create a Progress program to launch MFG/PRO using the CIM file.

Operation

- 1 Start your database servers, including those for the audit databases.
- 2 Start one or more background clients to flush the audit staging table in MFG/PRO.

Create a Batch Client Startup Script

Use the following steps to copy and update the batch client startup script.

- 1 Locate the following script in your *MFGPROInstallDir* and make a copy of it. The name of the copied script is a suggested name and is used for convenience in these instructions.

On UNIX systems, the file extension is `.ksh`; on Windows, `.bat`.

| | |
|--------------------------------|---------------------------------|
| Existing Script: | Copied Script: |
| <code>client.Production</code> | <code>client.ECBatchProd</code> |

- 2 Open `client.ECBatchProd` in a text editor. Locate the Progress executable, and make the following changes:
 - a Change the Progress executable name for the Progress client from `mpro` or `_progres` in UNIX, and from `mpro` or `_progres.exe` in Windows, to `mbpro`, the Progress batch mode executable.

Alternatively, you can add the `-b` parameter after the Progress executable, as in:

```
mpro -b
```

- b Replace `mf.p` with the batch launch program name, `ttpb.p`. This program runs MFG/PRO using a CIM file as input.

Note The instructions for writing the CIM file and the batch launch program follow this section.

The result, in a Windows script, looks like the following with changes shown in **bold**.

```
REM loop-db-start
%DLC%\bin\_progres -b -c 30 -d mdy -yy 1920 -Bt 350 -D 100 -
mmax 3000 -nb 200 -s 63 -noshvarfix -p ttpb.p -pf
c:\mfgsvr\Production.pf -ininame progress.chr
REM loop-db-end
pause
```

3 Save the file.

Create a CIM File

▶ For details on CIM, see *User Guide Volume 9: Manager Functions*.

CIM files are text files that MFG/PRO reads, treating each entry as an interface command. The CIM file to launch Audit Trail Creation Process should be named `ttp.cim`, and should contain a single line for each entry in the following table, in the exact order shown.

Create the file in a text editor and save it as `ttp.cim` in your `MFGPROInstallDir`.

Table 2.3
Enhanced Controls
CIM File Example

| CIM Entry | Descriptions |
|---------------|--|
| fr33 Na66Sand | Specifies a valid MFG/PRO log-in ID (fr33) and password (Na66Sand). |
| atttpui | Launches Audit Trail Creation Process (<code>atttpui.p</code>); the <code>.p</code> extension can be left off in the CIM file. |
| - | A blank line or dash (-) accepts the default log file. For CIM to accept the default, the default value must be populated in User Accountability Control (36.12.13.24). Alternatively, a fully qualified file path can be entered. |
| Yes | Provides a response to the message, “Please confirm start of Audit Trail Creation process.” |
| . | Reproduces an End key to complete or exit a program component. |
| . | Same as above, returning the program to the menu level. |
| N | No response to the MFG/PRO exit prompt. |
| . | End key to exit the MFG/PRO session. |
| Y | Confirms the exit from MFG/PRO. |

Create a Progress Batch Launch File

The batch launch program is a simple Progress program that loads the `ttp.cim` file, sends program output to a specified log file, starts an MFG/PRO client session, and then closes itself down. To use the batch launch program, follow these steps:

- 1 Open the Progress Editor. One method is to run the editor from the File menu in MFG/UTIL.
- 2 Enter the following program:

```
input from ttp.cim.  
output to ttpb.log.  
run mf.p.  
input close.  
output close.
```
- 3 Save this program as `ttpb.p` in your *MFGPROInstallDir*.

Running the Audit Process

- 1 Be sure that your audit database servers are running. If not, use the following script to start them:

```
start.EnhancedControls
```
- 2 Be sure that your MFG/PRO database servers are running. If they are not included in the Enhanced Controls script, use the following script to start them:

```
start.Production
```
- 3 Start one or more background clients to move audit data from the staging table in MFG/PRO:

```
client.ECBatchProd
```


Audit Database Sizing

This chapter includes reference information related to MFG/PRO audit databases required by the Enhanced Controls module.

| | |
|--------------------------------------|-----------|
| <i>Introduction</i> | 40 |
| <i>Audit Database Default Values</i> | 41 |
| <i>Progress Tools</i> | 41 |
| <i>MFG/PRO Database</i> | 42 |
| <i>Storage Area Assignments</i> | 45 |
| <i>Records per Block</i> | 46 |
| <i>Extents</i> | 48 |

Introduction

An audit database can be configured to store only audit trail data, only electronic signature archive data, or both. Guidelines are provided in this section for setting up storage areas and extents for audit databases.

However, audit databases will vary greatly in size depending on their configured use, the volume of changes being audited, and the volume of electronic signature data being archived. In addition, the database sizing guidelines in this section are based on data from a test environment and may not apply to your company's data. Live database configurations should be determined based on site-specific analyses.

When configuring a Progress database, the following considerations must be made and reflected in the data definition file (.df) and structure description file (.st) for the database:

- **Block Size:** Each Progress database is created with a single block size of 1, 2, 4, or 8K. This means that all memory blocks in the database are the same size. By default, MFG/PRO databases are created with an 8K block size.
- **Storage Area Assignments:** Storage areas give you physical control over the location of specific database objects. Storage areas can contain any combination of tables and indexes, but you cannot split a table or index across storage areas. Each table and index may be assigned to only one storage area.
- **Records per Block:** Each storage area in the database has a specific number of records per block. Storage areas can have 1, 2, 4, 8, 16, 32, 64, 128, or 256 records per block. The Progress default number of records per block is 64 if the block size is 8K, and 32 for all other block sizes. The records per block value can be chosen to maximize storage to disk (and minimize empty storage) by taking into account the mean record size to be stored in the storage area.
- **Extents:** An extent corresponds to a single OS file of a fixed or variable size. Each storage area can be split into one or more extents that allow you to extend storage areas across multiple files on the same or different physical volumes. The number and size of these extents must accommodate the data that will be stored during the lifetime of the database.

Audit Database Default Values

By default, all MFG/PRO databases are created with an 8K block size.

Based on actual audit trail data from a test environment, QAD has provided default storage area assignments and records per block values for those storage areas. These default values should work well for most companies, but if your company has chosen a different block size, then the default records per block values will need to be adjusted. The calculations of these default values are described in sections that follow.

The default extent sizes provided by QAD are not intended to meet the audit database storage needs of all companies and must be adjusted for each installation based on analysis of actual data. The default configuration provided for the audit database requires a total of 4GB for the fixed extents and is intended to be used only in a trial situation to determine the actual extents needed. This process is described in “Extents” on page 48.

Progress Tools

Database Record Analysis

The Progress `dbanalys` tool that is a part of the `proutil` utility can be used to analyze existing data in a database. As discussed in the sections that follow, this data is useful for determining the records per block and extent sizes. Run `dbanalys` from the Progress install `\bin` directory as follows:

```
proutil \mfgsvr\db\audprod.db -C dbanalys > out.txt
```

Go to the Record Block Summary portion of the report to view a breakdown of each storage area in the selected database. This summary shows the number of records for each table in the storage area, as well as the size in bytes, and the minimum, maximum, and mean record size in bytes. The Index Block Summary follows the record summary and shows similar information by index.

Storage Area Utilization

To monitor the free space in storage areas, the following sample 4GL code can be executed from a Progress Editor session connected to the database of interest:

```

for each _areastatus no-lock where _areastatus-areanum > 5:
  display
    _areastatus-areaname format "x(14)" label "Area"
    _areastatus-totblocks format ">>>,>>>,>>>" label "Total"
    _areastatus-hiwater format ">>>,>>>,>>>" label "Hi Water"
    _areastatus-totblocks - _areastatus-hiwater
      format ">>>,>>>,>>>" label "Free"
    _areastatus-hiwater / _areastatus-totblocks * 100
      format ">>9%" label "%Full".
end.

```

For an audit database, this will produce a display similar to the following:

| Area | Total | Hi Water | Free | %Full |
|-------------|---------|----------|---------|-------|
| ----- | ----- | ----- | ----- | ----- |
| Schema Area | 207 | 203 | 4 | 98% |
| ATMSTR | 32,002 | 15,702 | 16,300 | 49% |
| ATMSTR_IDX1 | 16,002 | 2406 | 13,596 | 15% |
| ATMSTR_IDX2 | 16,002 | 1011 | 14,991 | 6% |
| ATMSTR_IDX3 | 16,002 | 2605 | 13,397 | 16% |
| ATFDET | 250,002 | 113,008 | 136,994 | 45% |
| ATFDET_IDX1 | 16,002 | 2402 | 13,600 | 15% |
| ATFDET_IDX2 | 16,002 | 2709 | 13,293 | 17% |
| ATKDET | 32,002 | 5,407 | 26,595 | 17% |
| ATKDET_IDX1 | 16,002 | 1304 | 14,698 | 8% |
| ATKDET_IDX2 | 16,002 | 513 | 15,489 | 3% |
| ATKDET_IDX3 | 16,002 | 2603 | 13,399 | 16% |
| AESIG | 32,002 | 3 | 31,331 | 0% |
| AESIG_IDX | 32,002 | 14 | 31,988 | 0% |

MFG/PRO Database

Although the topic of this chapter is the audit database, this section describes the database configuration changes to one of the standard MFG/PRO databases. New storage areas have been designated in the `mfgprod` database for Enhanced Controls. These storage areas are not intended for long-term storage of this data, and so the sizing considerations are somewhat different than for the audit database.

Audit Trails

A new storage area, ATTMPMSTR, has been designated in the `mfgprod` database for the staging table that receives audit trail data during production activities. This data is then flushed to the audit database and deleted from the `attmp_mstr` staging table by one or more Audit Trail Creation Processes running in the background. The indexes for the staging table have been assigned to a new `ATTMPMSTR_IDX` storage area. Sizing for these storage areas should be of minor concern. The storage requirements will vary depending on the number of audited tables, frequency of changes to those tables, and number of background processes running. In general, you can control the required size of these storage areas by managing the number of background processes launched.

Start with one process per fifty users and monitor the table size throughout normal production activities. If you observe steadily increasing table size, increase the number of background processes accordingly.

Important If the background processes are shut down for some reason or if the audit database is not online, records remain in the staging table and could overrun the storage area assigned. Be sure to monitor the free space in these storage areas on a regular basis as described in “Progress Tools” on page 41.

The new storage area definitions in the default `mfgprod.st` structure file do not specify fixed extents and only specify one variable extent for each as shown below:

```
#
d "ATTMPMSTR" : 39,64 ./db
#
d "ATTMPMSTR_IDX" : 40,32 ./db
#
```

Note The structure file syntax above shows a comment sign (#) followed by the definition of one 64-record-per-block variable extent for the ATTMPMSTR storage area and is located in the `db` subdirectory. A variable extent can reach a maximum of 2GB in size.

These storage area definitions will need to be modified if audit trails will be enabled. Add one or more fixed extents to be consistent with the considerations described above and your sizing strategy for this database.

◆ See “Extents” on page 48 for additional information on sizing extents.

The sample below has been modified to provide one fixed extent (f) with a size of 5MB (5120) in addition to the variable extent for each storage area:

```
#
d "ATTMPMSTR":39,64 ./db f 5120
d "ATTMPMSTR":39,64 ./db
#
d "ATTMPMSTR_IDX":40,32 ./db f 5120
d "ATTMPMSTR_IDX":40,32 ./db
#
```

Electronic Signatures

A new ESIG storage area has been designated for the electronic signature transaction tables that have been added to the `mfgprod` database. The corresponding indexes have been assigned to a new `ESIG_IDX` storage area. Sizing for these storage areas should be based on your anticipated use of electronic signatures. The storage requirements will vary depending on the number of electronic signature-enabled programs, level of activity on those programs, and your archive/delete schedule for electronic signature data.

E-Signature Archive/Delete (36.12.14.22) can be used to archive electronic signature data to an audit database. Unlike for audit trails, the data is not automatically transferred to the audit database by a background process. During the archive/delete process, not all signature data will be deleted even if the data meets the specified date range. The latest signature for any signed record in MFG/PRO will not be deleted because that signature information must be available for display whenever the record is displayed or modified.

Important Be sure to monitor the free space in these storage areas on a regular basis as described in “Progress Tools” on page 41 and run the archive/delete menu program as needed.

The new storage area definitions in the default `mfgprod.st` structure file do not specify fixed extents and only specify one variable extent for each as shown below:

```
#
d "ESIG":41,64 ./db
#
d "ESIG_IDX":42,32 ./db
#
```

These storage area definitions will need to be modified if electronic signatures will be enabled. Add one or more fixed extents to be consistent with the considerations described above and your sizing strategy for this database. The sample below has been modified to provide one fixed extent (f) with a size of 5MB (5120) in addition to the variable extent for each storage area:

```
#
d "ESIG":41,64 ./db f 5120
d "ESIG":41,64 ./db
#
d "ESIG_IDX":42,32 ./db f 5120
d "ESIG_IDX":42,32 ./db
#
```

▶ See “Extents” on page 48 for additional information on sizing extents.

The remainder of this section focuses on the sizing considerations of the audit database.

Storage Area Assignments

See the Progress *Database Administration Guide and Reference* for detailed background on storage areas.

Audit Database

The audit database `audprod` contains three audit trail-related tables—`atf_det`, `atk_det`, and `at_mstr`—and eight indexes. By default, each of these tables and indexes is partitioned into its own new storage area for maximum storage capacity and flexibility.

The database also includes four electronic signature archive tables—`aescd_det`, `aescx_det`, `aesig_mstr`, and `aesrec_det`—along with ten indexes. Because the electronic signature volume will be relatively low, these four tables are assigned to one storage area and the corresponding indexes to a second storage area.

Table 3.1 shows the default assignment of each table and index to a storage area.

| Object Type | Used For | Object | Storage Area |
|-------------|-------------|--------------------------|--------------------------|
| Table | Audit Trail | <code>atf_det</code> | <code>ATFDET</code> |
| Index | Audit Trail | <code>oid_atf_det</code> | <code>ATFDET_IDX1</code> |

Table 3.1
Audit Database
Storage Areas

| Object Type | Used For | Object | Storage Area |
|-------------|---------------|----------------------|--------------|
| Index | Audit Trail | atf_oid_at_mstr | ATFDET_IDX2 |
| Table | Audit Trail | atk_det | ATKDET |
| Index | Audit Trail | oid_atk_det | ATKDET_IDX1 |
| Index | Audit Trail | atk_field_name | ATKDET_IDX2 |
| Index | Audit Trail | atk_oid_at_mstr | ATKDET_IDX3 |
| Table | Audit Trail | at_mstr | ATMSTR |
| Index | Audit Trail | oid_at_mstr | ATMSTR_IDX1 |
| Index | Audit Trail | at_table_name | ATMSTR_IDX2 |
| Index | Audit Trail | at_oid_erecord | ATMSTR_IDX3 |
| Table | E-Sig Archive | aescd_det | AESIG |
| Index | E-Sig Archive | aescd_oid_escat_mstr | AESIG_IDX |
| Index | E-Sig Archive | oid_aescd_det | AESIG_IDX |
| Table | E-Sig Archive | aescx_det | AESIG |
| Index | E-Sig Archive | aescx_oid_det | AESIG_IDX |
| Index | E-Sig Archive | oid_aescx_det | AESIG_IDX |
| Table | E-Sig Archive | aesig_mstr | AESIG |
| Index | E-Sig Archive | aesig_date | AESIG_IDX |
| Index | E-Sig Archive | aesig_userid | AESIG_IDX |
| Index | E-Sig Archive | oid_aesig_mstr | AESIG_IDX |
| Table | E-Sig Archive | aesrec_det | AESIG |
| Index | E-Sig Archive | aesrec_table_name | AESIG_IDX |
| Index | E-Sig Archive | aesrec_record_order | AESIG_IDX |
| Index | E-Sig Archive | oid_aesrec_det | AESIG_IDX |

Records per Block

General Recommendations

The default records per block values for the audit database storage areas were determined using the following analysis of actual audit database data from a test environment. These defaults should work well for most companies because the mean record size per table is not expected to vary much from the test data analyzed.

Note This information assumes the default 8K database block size. If you are using a different block size, you will need to adjust the records per block values in the `audprod.st` structure file. You can substitute other database block sizes in the formulas provided.

To configure storage areas to maximize storage to disk (and minimize empty storage), you need to know the mean record size you are writing to disk. A simple formula for estimating the records per block is:

$$\text{database block size} / [\text{mean record size} + 20 \text{ (estimated row overhead)}] = \text{records per block}$$

For example, if you are writing 1K records (mean size) to an 8K block size database (8192 bytes), you will want to create the storage area with 8 records per block.

$$8192 / (1024 + 20) = 8$$

When multiple tables are stored in the same storage area, the mean record size will need to be averaged over all tables as you will see in the specific examples. You will also not usually end up with a round number. Round the number up or down to the closest permissible records per block.

Audit Database

The audit database from a QAD test environment was analyzed using the `proutil` command and the output is summarized in Table 3.2.

| Table | #Records | Size | Record Size | | |
|------------|----------|--------|-------------|------|------|
| | | | Min | Max | Mean |
| atf_det | 134474 | 113.0M | 63 | 3073 | 881 |
| atk_det | 82497 | 5.4M | 54 | 94 | 69 |
| at_mstr | 134434 | 15.7M | 96 | 150 | 122 |
| aescd_det | 366 | 38.8K | 97 | 112 | 108 |
| aescx_det | 559 | 559.4K | 90 | 1506 | 1024 |
| aesig_mstr | 259 | 22.8K | 75 | 119 | 90 |
| aesrec_det | 533 | 49.3K | 82 | 99 | 94 |

Table 3.2
Record Size in Bytes for the Audit Database

This data was then used to calculate the default records per block for the `audprod.st` structure definition file. The calculation for the AESIG storage area is described here.

The total mean record size for a storage area is determined by adding 20 to each table's mean record size and averaging the means weighted by the number of records. For the AESIG storage area, the mean record size would be:

$$[128 (366) + 1044 (559) + 110 (259) + 114 (533)] / (366 + 559 + 259 + 533) = 419$$

The database block size (in bytes) is then divided by the total mean record size:

$$8192 / 419 = 19.6$$

This result is then rounded up or down to the nearest permissible records per block, which in this case is 16.

For storage areas that will store records for only one table, the total mean record size for the storage area will just be the mean record size for the table plus 20.

Table 3.3 shows the resulting records per block for the audit database.

Table 3.3
Number of Records
per Block for the
Audit Database

| Storage Area | Database Block Size/ Mean Record Size | Recommended Records/Block |
|--------------|--|------------------------------|
| ATFDET | 8192 / 901 = 9.1 | 8 |
| ATKDET | 8192 / 89 = 92 | 128 |
| ATMSTR | 8192 / 142 = 58 | 64 |
| AESIG | 8192 / 419 = 19.6 | 16 |

The default records per block for storage areas containing only indexes is 32 for all MFG/PRO databases. This default should work well for the audit database also.

Extents

General Recommendations

In general, fewer extents are optimal for improved performance. The fewer the extents, the less system resources are needed (such as file handles), the less time it takes to open the database, and the easier it is to administer the database.

The default recommendation for each extent size is just under 2 GB (2000000 in the structure file). However, a safe extent size of 0.5 GB is also acceptable for most large databases (512000 in the structure file). If the database is smaller than 2 GB, use a smaller fixed extent with one variable extent.

For databases where performance is not critical, use one or a few large fixed extents. Always have at least one variable extent as a safeguard.

You should focus on achieving good input/output (I/O) performance. To do this, the database should be spread out across multiple disk drives. The more drives, the greater the overall system I/O capacity. Progress recommends that you use striped disk arrays and mirrored disks, and that you isolate the before-image and after-image files.

▶ See the Progress documentation for more information.

There is a maximum size for a storage area that is independent of the number of extents but dependent on the block size and records per block as shown in Table 3.4 for an 8K block size.

▶ See the Progress documentation for the limits for other block sizes.

| Database Block Size | Records Per Block | Maximum Area Size |
|---------------------|-------------------|-------------------|
| 8192 bytes (8K) | 1 | 16TB |
| 8192 bytes (8K) | 2 | 8TB |
| 8192 bytes (8K) | 4 | 4TB |
| 8192 bytes (8K) | 8 | 2TB |
| 8192 bytes (8K) | 16 | 1TB |
| 8192 bytes (8K) | 32 | 512GB |
| 8192 bytes (8K) | 64 (default) | 256GB |
| 8192 bytes (8K) | 128 | 128GB |
| 8192 bytes (8K) | 256 | 64GB |

Table 3.4
Progress Maximum Storage Area Sizes

Audit Database

The best way to determine the size of the storage area extents is to actually audit trail a live system for a short period of time and then analyze the resulting audit database. Alternatively, this can be done in a test environment as long as the activity of the live system can be simulated in the test environment. Create an audit database to use for the trial period using the provided default structure file, `audprod.st`. This database will require a total of 4 GB for the fixed extents. After

determining the tables of interest to be audit trailed, configure audit trail profiles for those tables to begin on a particular day (Audit Trail = On). Audit trails will be generated starting at 12:01 AM. that day. Make sure to have an Audit Trail Creation Process running so that the audit trails will be transferred to the audit database. After running the system for the desired period of time, perhaps a few hours, one day, or even a few days, run the Progress dbanalys utility against the audit database, saving the output to a file.

Table 3.5
Sample Table
Results of Database
Analysis Report

| Table | #Records | Size | Record Size | | |
|------------|-----------|--------|-------------|------|------|
| | | | Min | Max | Mean |
| atf_det | 1,075,792 | 904.0M | 63 | 3073 | 881 |
| atk_det | 659,976 | 43.2M | 54 | 94 | 69 |
| at_mstr | 1,075,472 | 125.6M | 96 | 150 | 122 |
| aescd_det | 2,928 | 310.4K | 97 | 112 | 108 |
| aescx_det | 4,472 | 4.475M | 90 | 1506 | 1024 |
| aesig_mstr | 2,072 | 182.4K | 75 | 119 | 90 |
| aesrec_det | 4,264 | 394.4K | 82 | 99 | 94 |

Assuming that the numbers in Table 3.5 are from running audit trails in a live system for one day, you can estimate the required size for each storage area according to the life expectancy for the audit database.

Example If the audit database is to be active for 3 months (65 working days), the required storage area sizes would be calculated as shown in Table 3.6.

Table 3.6
Extent Sizing
Based on Sample
Analysis Report

| Storage Area | Tables Included | 1-Day Size | 65-Day Calculated Size |
|--------------|---|------------|------------------------|
| ATFDET | atf_det | 904.0M | 58.76G |
| ATKDET | atk_det | 43.2M | 2.808G |
| ATMSTR | at_mstr | 125.6M | 8.164G |
| AESIG | aescd_det, aescx_det, aesig_mstr, aesrec_det | 5.362M | 348.5M |

It is always possible and usually desirable to create multiple fixed extents to satisfy the calculated size of a single storage area. In fact, this would be required if the calculated size exceeds the maximum file limit of the operating system. Alternatively, the life expectancy of the database can be

reduced. For example, a new database can be brought online every two months instead of every three months. If the calculated size of a storage area exceeds the Progress maximum size as listed in Table 3.4 then the life expectancy of the database must be reduced.

If a fixed extent is to have a size of 2 GB or larger, be sure to enable large files for the database using the following command:

```
proutil <db_name> -C EnableLargeFiles
```

The operating system also needs to have large files enabled. Refer to your operating system documentation for further information.

As a safeguard, always have at least one variable extent in addition to one or more fixed extents for each storage area.

Note Because a variable extent can grow to 2 GB if supported and enabled in the operating system, it is a good idea to enable large files in the database even if the fixed extents are less than 2 GB.

A similar analysis of the index sizes can be used to determine the extent sizes for those storage areas as described below.

| Table | Index | Size |
|------------|----------------------|-------|
| atf_det | atf_oid_at_mstr | 21.6M |
| | oid_atf_det | 19.2M |
| atk_det | atk_field_name | 4.08M |
| | atk_oid_at_mstr | 20.8M |
| | oid_atk_det | 10.4M |
| at_mstr | at_oid_erecord | 20.8M |
| | at_table_name | 8.00M |
| | oid_at_mstr | 19.2M |
| aescd_det | aescd_oid_escat_mstr | 48.8K |
| | oid_aescd_det | 43.2K |
| aescx_det | aescx_oid_aescd_det | 64.0K |
| | oid_aescx_det | 57.6K |
| aesig_mstr | aesig_date | 21.6K |
| | aesig_userid | 23.2K |
| | oid_aesig_mstr | 31.2K |

Table 3.7
Sample Index
Results of Database
Analysis Report

| Table | Index | Size |
|------------|---------------------|-------|
| aesrec_det | aesrec_record_order | 60.8K |
| | aesrec_table_name | 36.0K |
| | oid_aesrec_det | 62.4K |

Assuming that the numbers in Table 3.7 are the dbanalyis output for the one day of system use, the required extents would be calculated as shown in Table 3.8.

Table 3.8
Index Extent Sizing
Based on Sample
Analysis Report

| Storage Area | Indexes Included | 1-Day Size | 65-Day Calculated Size |
|--------------|--|------------|------------------------|
| ATFDET_IDX1 | oid_atf_det | 19.2M | 1.25G |
| ATFDET_IDX2 | atf_oid_at_mstr | 21.6M | 1.40G |
| ATKDET_IDX1 | oid_atk_det | 10.4M | 676M |
| ATKDET_IDX2 | atk_field_name | 4.08M | 265M |
| ATKDET_IDX3 | atk_oid_at_mstr | 20.8M | 1.35G |
| ATMSTR_IDX1 | oid_at_mstr | 19.2M | 1.25G |
| ATMSTR_IDX2 | at_table_name | 8.00M | 520M |
| ATMSTR_IDX3 | at_oid_erecord | 20.8M | 1.35G |
| AESIG_IDX | aescd_oid_escat_mstr, oid_aescd_det, aescx_oid_det, oid_aescx_det, aesig_date, aesig_userid, oid_aesig_mstr, aesrec_table_name, aesrec_record_order, oid_aesrec_det | 449K | 29.2M |

Index

A

- audit databases
 - administering 7
 - naming 9
 - sizing guidelines 39
- Audit Trail Creation Process 34
- auditing 7
 - storage area 43

B

- batch
 - startup for Enhanced Controls 34

C

- CIM file
 - for batch startup 36
- client/server
 - services file 12
- compile
 - additional languages 34
 - application code 29
 - compiler options 30
 - multithreaded 29

D

- database server installation
 - services file 12
 - working databases 20
- databases
 - working 20
- dbanalys tool 41
- document conventions 3
- documents
 - related 3

E

- Enable Enhanced Controls Schema Changes 18

Enhanced Controls

- batch startup 34
- Enhanced Controls installation 11–37
- E-Signature Archive/Delete 44
- e-signatures
 - loading data 22
 - overview 6
 - storage areas 44

I

- installation
 - planning 5
- installation summary 6

M

- multi-language
 - compiles for 34
- multithreaded compiles 29

O

- object IDs (OIDs) 10
- OIDs 10
 - populating 17

P

- PROPATH
 - compile 26, 31

Q

- QAD Global Support 2, 4

S

- scripts
 - client startup 26
 - generating 26
 - server startup 26
- services file 12

storage areas

- ATTMPMSTR 43
- ATTMPMSTR_IDX 43
- ESIG 44
- ESIG_IDX 44

structure files

- audit database 13

W

Web site

- QAD Support 2
- wk0060.ini 9
- wk0310.ini 10, 15

workflows

- Create Production Progress Audit Databases 10, 15
- Enable Enhanced Controls Schema Changes 9, 18
- working databases 20