

Progress Basic Database Tools

© 2001 Progress Software Corporation. All rights reserved.

Progress® software products are copyrighted and all rights are reserved by Progress Software Corporation. This manual is also copyrighted and all rights are reserved. This manual may not, in whole or in part, be copied, photocopied, translated, or reduced to any electronic medium or machine-readable form without prior consent, in writing, from Progress Software Corporation.

The information in this manual is subject to change without notice, and Progress Software Corporation assumes no responsibility for any errors that may appear in this document.

The references in this manual to specific platforms supported are subject to change.

Progress, Progress Results, Provision and WebSpeed are registered trademarks of Progress Software Corporation in the United States and other countries. Apptivity, AppServer, ProVision Plus, SmartObjects, IntelliStream, and other Progress product names are trademarks of Progress Software Corporation.

SonicMQ is a trademark of Sonic Software Corporation in the United States and other countries.

Progress Software Corporation acknowledges the use of Raster Imaging Technology copyrighted by Snowbound Software 1993-1997 and the IBM XML Parser for Java Edition.

© IBM Corporation 1998-1999. All rights reserved. U.S. Government Users Restricted Rights — Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Progress is a registered trademark of Progress Software Corporation and is used by IBM Corporation in the mark Progress/400 under license. Progress/400 AND 400® are trademarks of IBM Corporation and are used by Progress Software Corporation under license.

Java and all Java-based marks are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

Any other trademarks and/or service marks contained herein are the property of their respective owners.

.

May 2001



Product Code: 4504
Item Number: 81074; 9.1C

Contents

Preface . . . . . **xi**
Purpose xi
Audience xi
Organization of This Manual xi
Typographical Conventions xii
Syntax Notation xiii
Progress Messages xvii
Other Useful Documentation xviii
Getting Started xviii
Development Tools xx
Reporting Tools xxi
4GL xxi
Database xxii
DataServers xxii
SQL-89/Open Access xxiii
SQL-92 xxiii
Deployment xxiv
WebSpeed xxiv
Reference xxv

1. Database Tasks **1–1**
1.1 Starting the Data Dictionary 1–2
1.2 Database Tasks 1–3
1.2.1 Creating a Database 1–3
1.2.2 Connecting a Database 1–4
1.2.3 Switching to Another Connected Database 1–5
1.2.4 Disconnecting a Database 1–5
1.2.5 Viewing Session Information 1–5
1.2.6 Dumping Data Definitions 1–6

1.2.7 Creating an Incremental Data Definition File 1–6
1.2.8 Creating a Parameter File 1–7
1.2.9 Editing a Parameter File 1–7
1.2.10 Loading Data Definitions 1–8

2. Table Tasks 2–1
2.1 Creating a Table 2–2
2.1.1 Specifying Validation Criteria 2–2
2.1.2 Specifying String Attributes 2–3
2.2 Modifying a Table 2–3
2.3 Deleting a Table 2–4
2.4 Freezing/Unfreezing a Table 2–4
2.5 Generating a Table Report 2–5
2.5.1 Generating a Detailed Table Report 2–5
2.5.2 Generating a Quick Table Report 2–6
2.5.3 Generating a Table Relations Report 2–6
2.6 Dumping Table Contents 2–7
2.6.1 Dumping User Table Contents 2–8
2.7 Loading Table Contents 2–9
2.7.1 Loading User Table Contents 2–10
2.7.2 Reconstructing Bad Load Records 2–10
2.8 Binary Dump and Load 2–11
2.8.1 Performing the Binary Load 2–13
2.9 Creating a Bulk Loader Description File 2–13
2.10 Generating an Include File 2–14
2.10.1 Generating an ASSIGN Statement Include File 2–14
2.10.2 Generating a FORM Statement Include File 2–14
2.10.3 Generating a DEFINE WORK-TABLE Statement Include File . 2–15

3. Field Tasks 3–1
3.1 Adding a Field 3–2
3.2 Modifying a Field 3–3
3.3 Changing a Field Name Globally 3–4
3.4 Copying a Field 3–4
3.5 Deleting a Field 3–5
3.6 Changing Field Order 3–6
3.7 Generating a Quick Field Report 3–6
3.8 Adjusting the SQL Field Width 3–7

4. Index Tasks 4–1
4.1 Creating an Index 4–2
4.2 Modifying an Index 4–4
4.3 Changing the Primary Index for a Table 4–4
4.4 Renaming an Index 4–4

4.5 Deactivating an Index 4–5
4.6 Deleting an Index 4–5
4.7 Generating a Quick Index Report 4–6

5. Sequence Tasks 5–1
5.1 Creating a Sequence 5–2
5.2 Modifying a Sequence 5–3
5.3 Deleting a Sequence 5–3
5.4 Dumping Sequence Definitions 5–4
5.5 Dumping Sequence Current Values 5–4
5.6 Loading Sequence Current Values 5–5
5.7 Generating a Sequence Report 5–5

6. Trigger Tasks 6–1
6.1 Creating a Trigger 6–2
6.1.1 Creating Table Triggers 6–2
6.1.2 Creating a Field Trigger 6–3
6.2 Modifying a Trigger 6–4
6.2.1 Modifying a Table Trigger 6–4
6.2.2 Modifying a Field Trigger 6–4
6.3 Deleting a Trigger 6–5
6.3.1 Deleting a Table Trigger 6–5
6.3.2 Deleting a Field Trigger 6–5
6.4 Generating a Trigger Report 6–6

7. SQL-89 Tasks 7–1
7.1 Generating an SQL-89 Program 7–2
7.1.1 Generating CREATE VIEW Statements 7–2
7.1.2 Generating CREATE TABLE Statements 7–2
7.2 Dumping the Definitions for SQL-89 Views 7–3
7.3 Loading the Definitions for SQL-89 Views 7–3
7.4 Generating an SQL-89 View Report 7–3

8. Database Menu Reference 8–1
8.1 Data Dictionary Menu Bar 8–2
8.2 Database Menu 8–3
8.2.1 Database→ Select Working Database. 8–4
8.2.2 Database→ Create 8–5
8.2.3 Database→ Connect 8–7
8.2.4 Database→ Disconnect 8–9
8.2.5 Database→ Reports. 8–9
8.2.6 Database→ Exit 8–19

9. Schema Menu Reference 9–1
9.1 Schema Menu 9–2
9.1.1 Schema→ Modify Table 9–3
9.1.2 Schema→ Add New Table 9–11
9.1.3 Schema→ Delete Table(s) 9–11
9.1.4 Schema→ Field Editor 9–11
9.1.5 Schema→ Reorder Fields 9–30
9.1.6 Schema→ Global Field Name Change 9–31
9.1.7 Schema→ Index Editor 9–31
9.1.8 Schema→ Sequence Editor 9–36

10. Admin Menu Reference 10–1
10.1 Admin Menu 10–2
10.1.1 Admin→ Dump Data and Definitions 10–2
10.1.2 Admin→ Load Data and Definitions 10–7
10.1.3 Admin→ Security 10–10
10.1.4 Admin→ Export Data 10–16
10.1.5 Admin→ Import Data 10–19

11. Advanced Menus Reference 11–1
11.1 DataServer Menu 11–2
11.2 Utilities Menu 11–2
11.2.1 Utilities→ Editor for Parameter Files 11–3
11.2.2 Utilities→ Quoter Functions 11–4
11.2.3 Utilities→ Generate Include Files 11–7
11.2.4 Utilities→ Edit Progress Auto-Connect List 11–11
11.2.5 Utilities→ Freeze/Unfreeze Table 11–13
11.2.6 Utilities→ Index Deactivation 11–14
11.2.7 Utilities→ Information 11–14
11.3 PRO/SQL Menu 11–15
11.3.1 PRO/SQL→ PRO/SQL View Report 11–16
11.3.2 PRO/SQL→ Dump as CREATE VIEW Statement 11–18
11.3.3 PRO/SQL→ Dump as CREATE TABLE Statements 11–19
11.4 Tools Menu 11–20

Index Index–1

Figures

Figure 1–1: Data Dictionary Window 1–2
Figure 8–1: Data Dictionary Menu Bar 8–2
Figure 8–2: Select Working Database Dialog Box 8–4
Figure 8–3: Create Database Dialog Box 8–5
Figure 8–4: Find Database File Dialog Box 8–6
Figure 8–5: Database Connect Dialog Box 8–7
Figure 8–6: Report Options Dialog Box 8–10
Figure 8–7: Detailed Table Report Dialog Box 8–11
Figure 8–8: Quick Table Report Dialog Box 8–12
Figure 8–9: Quick Field Report Dialog Box 8–13
Figure 8–10: Quick Index Report Dialog Box 8–14
Figure 8–11: PRO/SQL View Report Dialog Box 8–15
Figure 8–12: Sequence Report Dialog Box 8–15
Figure 8–13: Trigger Report Dialog Box 8–16
Figure 8–14: User Report Dialog Box 8–17
Figure 8–15: Table Relations Report Dialog Box 8–18
Figure 8–16: Quick Storage Areas Report Dialog Box 8–19
Figure 9–1: Modify Table Window 9–3
Figure 9–2: Table Validation Dialog Box 9–5
Figure 9–3: Table Triggers Dialog Box 9–6
Figure 9–4: Trigger Code Dialog Box 9–8
Figure 9–5: Table String Attributes Dialog Box 9–8
Figure 9–6: Add New Table Window 9–11
Figure 9–7: Field Editor Window 9–12
Figure 9–8: Field Add Option Window 9–14
Figure 9–9: Field Triggers Dialog Box 9–27
Figure 9–10: View-As Phrase Dialog Box 9–28
Figure 9–11: GoIndex Option Window 9–30
Figure 9–12: Resequence Order of Fields Dialog Box 9–30
Figure 9–13: Global Field Name Change Dialog Box 9–31
Figure 9–14: Index Editor Window 9–32
Figure 9–15: Index Add Option Window 9–35
Figure 9–16: Sequence Editor Window 9–36
Figure 10–1: Dump Data Contents for Some Tables Dialog Box 10–4
Figure 10–2: Create Incremental .df File Window 10–6
Figure 10–3: Reconstruct Bad Load Records Dialog Box 10–9
Figure 10–4: Edit User List Window 10–11
Figure 10–5: Edit Data Security Dialog Box 10–13
Figure 10–6: Security Administrators Dialog Box 10–15
Figure 10–7: User Report Dialog Box 10–16
Figure 10–8: Export DIF Dialog Box 10–17
Figure 10–9: Output Record Formatting Dialog Box 10–19
Figure 10–10: Import DIF Dialog Box 10–20

Figure 10–11: Import Fixed-Length Dialog Box 10–21
Figure 10–12: Fixed Length Field Columns Dialog Box 10–22
Figure 10–13: Import dBASE II, III, III+ and IV Definitions Dialog Box 10–22
Figure 11–1: Edit Parameter File Dialog Box 11–3
Figure 11–2: Parameter File Comments Dialog Box 11–4
Figure 11–3: Quote Entire Lines Dialog Box 11–5
Figure 11–4: Quote By Delimiter Dialog Box 11–5
Figure 11–5: Quote By Column Ranges Dialog Box 11–6
Figure 11–6: ASSIGN Statement Example 11–8
Figure 11–7: Generate FORM Dialog Box 11–8
Figure 11–8: FORM Statement Example 11–10
Figure 11–9: DEFINE WORK-TABLE Statement Example 11–11
Figure 11–10: Edit Progress Auto-Connect List Window 11–12
Figure 11–11: Freeze/Unfreeze Table Dialog Box 11–13
Figure 11–12: Index Deactivation Dialog Box 11–14
Figure 11–13: Session Information Dialog Box 11–15
Figure 11–14: View Report 11–16
Figure 11–15: Dump CREATE VIEW Dialog Box 11–18
Figure 11–16: Dump CREATE TABLE Dialog Box 11–19

Tables

Table 8–1: Data Dictionary Menu 8–2
Table 8–2: Options Available Without a Connected Database 8–3
Table 8–3: Database Menu 8–4
Table 8–4: Reports Menu 8–9
Table 9–1: Schema Menu 9–2
Table 9–2: Table Trigger Events 9–7
Table 9–3: String Attribute Characters 9–9
Table 9–4: String Attribute Examples 9–10
Table 9–5: Field Editor Options 9–12
Table 9–6: Character Format Symbols 9–15
Table 9–7: Character Display Format Examples 9–16
Table 9–8: Decimal and Integer Format Characters 9–16
Table 9–9: Numeric Display Format Examples 9–18
Table 9–10: Date Display Format Examples 9–20
Table 9–11: Logical Display Format Examples 9–20
Table 9–12: Field Label Examples 9–21
Table 9–13: Field Initial Values 9–22
Table 9–14: Data Types 9–23
Table 9–15: View As Widgets 9–28
Table 9–16: Index Editor Options 9–34
Table 9–17: Sequence Editor Options 9–38
Table 10–1: Admin Menu 10–2
Table 10–2: Dump Data and Definitions Menu 10–2
Table 10–3: Load Data and Definitions Menu 10–7
Table 10–4: Security Menu 10–10
Table 10–5: Edit User List Window Options 10–11
Table 10–6: Access Restrictions for Tables 10–14
Table 10–7: Edit Data Security Dialog Box Options 10–14
Table 10–8: Export Data Menu 10–16
Table 10–9: Export Format File Extensions 10–18
Table 10–10: Import Data Menu 10–20
Table 11–1: DataServer Menu 11–2
Table 11–2: Utilities Menu 11–2
Table 11–3: Quoter Functions Menu 11–4
Table 11–4: Generate Include Files Menu 11–7
Table 11–5: Edit Auto-Connect List Options 11–13
Table 11–6: PRO/SQL Menu 11–15

Preface

Purpose

This book is a user guide for the Progress Version 9 basic database administration toolset for character-mode.

Audience

This book is intended for developers who want to use the Progress character-mode Data Dictionary to manage the Progress database and their applications.

Organization of This Manual

Chapter 1, “Database Tasks”

Describes how to access the Data Dictionary and how to use it to perform database tasks.

Chapter 2, “Table Tasks”

Describes how to perform administrative tasks for tables and files.

Chapter 3, “Field Tasks”

Describes how to perform tasks to create and maintain fields.

Chapter 4, “Index Tasks”

Describes how to perform tasks to create and maintain indexes.

Chapter 5, “Sequence Tasks”

Describes how to perform tasks to create and maintain sequences.

Chapter 6, “Trigger Tasks”

Describes how to perform tasks to create and maintain schema triggers.

Chapter 7, “SQL-89 Tasks”

Describes how to perform SQL tasks including dumping and loading definitions for SQL views.

Chapter 8, “Database Menu Reference”

Describes the Database menu options and dialog boxes.

Chapter 9, “Schema Menu Reference”

Describes the Schema menu options and dialog boxes.

Chapter 10, “Admin Menu Reference”

Describes the Admin menu options and dialog boxes.

Chapter 11, “Advanced Menus Reference”

Describes the menu options and dialog boxes for the Utilities and PRO/SQL menus.

Typographical Conventions

This manual uses the following typographical conventions:

- **Bold typeface** indicates:
  - Commands or characters that the user types
  - That a word carries particular weight or emphasis
- *Italic typeface* indicates:
  - Progress variable information that the user supplies
  - New terms
  - Titles of complete publications
- `Monospaced typeface` indicates:
  - Code examples

- System output
- Operating system filenames and pathnames

The following typographical conventions are used to represent keystrokes:

- Small capitals are used for Progress key functions and generic keyboard keys.

  END-ERROR, **GET**, **GO**
  ALT, **CTRL**, **SPACEBAR**, **TAB**

- When you have to press a combination of keys, they are joined by a dash. You press and hold down the first key, then press the second key.

  CTRL-X

- When you have to press and release one key, then press another key, the key names are separated with a space.

  ESCAPE H
  ESCAPE CURSOR-LEFT

Syntax Notation

The syntax for each component follows a set of conventions:

- Uppercase words are keywords. Although they are always shown in uppercase, you can use either uppercase or lowercase when using them in a procedure.

  In this example, ACCUM is a keyword:

  SYNTAX

  ```
  ACCUM aggregate expression
  ```

- Italics identify options or arguments that you must supply. These options can be defined as part of the syntax or in a separate syntax identified by the name in italics. In the ACCUM function above, the *aggregate* and *expression* options are defined with the syntax for the ACCUM function in the *Progress Language Reference*.

- You must end all statements (except for DO, FOR, FUNCTION, PROCEDURE, and REPEAT) with a period. DO, FOR, FUNCTION, PROCEDURE, and REPEAT statements can end with either a period or a colon, as in this example:

```
FOR EACH Customer:
   DISPLAY Name.
END.
```

- Square brackets (**[ ]**) around an item indicate that the item, or a choice of one of the enclosed items, is optional.

  In this example, STREAM *stream*, UNLESS-HIDDEN, and NO-ERROR are optional:

  SYNTAX

```
DISPLAY [ STREAM stream ] [ UNLESS-HIDDEN ] [ NO-ERROR ]
```

  In some instances, square brackets are not a syntax notation, but part of the language.

  For example, this syntax for the INITIAL option uses brackets to bound an initial value list for an array variable definition. In these cases, normal text brackets ([ ]) are used:

  SYNTAX

```
INITIAL [ constant [ , constant ] ... ]
```

  NOTE: The ellipsis (**. . .**) indicates repetition, as shown in a following description.

- Braces (**{ }**) around an item indicate that the item, or a choice of one of the enclosed items, is required.

  In this example, you must specify the items BY and *expression* and can optionally specify the item DESCENDING, in that order:

  SYNTAX

```
{ BY expression [ DESCENDING ] }
```

In some cases, braces are not a syntax notation, but part of the language.

For example, a called external procedure must use braces when referencing arguments passed by a calling procedure. In these cases, normal text braces ({ }) are used:

SYNTAX

```
{ &argument-name }
```

- A vertical bar (**|**) indicates a choice.

  In this example, EACH, FIRST, and LAST are optional, but you can only choose one:

  SYNTAX

  ```
  PRESELECT [ EACH | FIRST | LAST ] record-phrase
  ```

  In this example, you must select one of *logical-name* or *alias*:

  SYNTAX

  ```
  CONNECTED ( { logical-name | alias } )
  ```

- Ellipses (**. . .**) indicate that you can choose one or more of the preceding items. If a group of items is enclosed in braces and followed by ellipses, you must choose one or more of those items. If a group of items is enclosed in brackets and followed by ellipses, you can optionally choose one or more of those items.

  In this example, you must include two expressions, but you can optionally include more. Note that each subsequent expression must be preceded by a comma:

  SYNTAX

  ```
  MAXIMUM ( expression , expression [ , expression ] ... )
  ```

In this example, you must specify MESSAGE, then at least one of *expression* or SKIP, but any additional number of *expression* or SKIP is allowed:

SYNTAX

```
MESSAGE { expression | SKIP [ (n) ] } ...
```

In this example, you must specify {*include-file*, then optionally any number of *argument* or &*argument-name* = "*argument-value*", and then terminate with }:

SYNTAX

```
{ include-file
    [ argument | &argument-name = "argument-value" ] ... }
```

- In some examples, the syntax is too long to place in one horizontal row. In such cases, **optional** items appear individually bracketed in multiple rows in order, left-to-right and top-to-bottom. This order generally applies, unless otherwise specified. **Required** items also appear on multiple rows in the required order, left-to-right and top-to-bottom. In cases where grouping and order might otherwise be ambiguous, braced (required) or bracketed (optional) groups clarify the groupings.

  In this example, WITH is followed by several optional items:

  SYNTAX

```
WITH [ ACCUM max-length ] [ expression DOWN ]
  [ CENTERED ] [ n COLUMNS ] [ SIDE-LABELS ]
  [ STREAM-IO ]
```

In this example, ASSIGN requires one of two choices: either one or more of *field*, or one of *record*. Other options available with either *field* or *record* are grouped with braces and brackets. The open and close braces indicate the required order of options:

SYNTAX

```
ASSIGN {   { [ FRAME frame ]
               { field [ = expression ] }
               [ WHEN expression ]
           } ...
       |  { record [ EXCEPT field ... ] }
       }
```

Progress Messages

Progress displays several types of messages to inform you of routine and unusual occurrences:

- Execution messages inform you of errors encountered while Progress is running a procedure (for example, if Progress cannot find a record with a specified index field value).
- Compile messages inform you of errors found while Progress is reading and analyzing a procedure prior to running it (for example, if a procedure references a table name that is not defined in the database).
- Startup messages inform you of unusual conditions detected while Progress is getting ready to execute (for example, if you entered an invalid startup parameter).

After displaying a message, Progress proceeds in one of several ways:

- Continues execution, subject to the error-processing actions that you specify, or that are assumed, as part of the procedure. This is the most common action taken following execution messages.
- Returns to the Progress Procedure Editor so that you can correct an error in a procedure. This is the usual action taken following compiler messages.
- Halts processing of a procedure and returns immediately to the Procedure Editor. This does not happen often.
- Terminates the current session.

Progress messages end with a message number in parentheses. In this example, the message number is 200:

```
** Unknown table name table. (200)
```

You can use the Progress PRO command to start a single-user mode character Progress client session and view a brief description of a message by providing its number. Follow these steps:

1 ♦ Start the Progress Procedure Editor:

```
install-dir/dlc/bin/pro
```

2 ♦ Press **F3** to access the menu bar, then choose Help→ Messages.

3 ♦ Type the message number, and press **ENTER**. Details about that message number appear.

4 ♦ Press **F4** to close the message, press **F3** to access the Procedure Editor menu, and choose File→ Exit.

Other Useful Documentation

This section lists Progress Software Corporation documentation that you might find useful. Unless otherwise specified, these manuals support both Windows and Character platforms and are provided in electronic documentation format on CD-ROM.

Getting Started

Progress Electronic Documentation Installation and Configuration Guide (Hard copy only)

A booklet that describes how to install the Progress EDOC viewer and collection on UNIX and Windows.

Progress Installation and Configuration Guide Version 9 for UNIX

A manual that describes how to install and set up Progress Version 9.1 for the UNIX operating system.

Progress Installation and Configuration Guide Version 9 for Windows

A manual that describes how to install and set up Progress Version 9.1 for all supported Windows and Citrix MetaFrame operating systems.

Progress Version 9 Product Update Bulletin

A guide that provides a brief description of each new feature of the release. The booklet also explains where to find more detailed information in the documentation set about each new feature.

Progress Application Development Environment — Getting Started (Windows only)

A practical guide to graphical application development within the Progress Application Development Environment (ADE). This guide includes an overview of the ADE and its tools, an overview of Progress SmartObject technology, and tutorials and exercises that help you better understand SmartObject technology and how to use the ADE to develop applications.

Progress Language Tutorial for Windows and *Progress Language Tutorial for Character*

Platform-specific tutorials designed for new Progress users. The tutorials use a step-by-step approach to explore the Progress application development environment using the 4GL.

Progress Master Glossary for Windows and *Progress Master Glossary for Character* (EDOC only)

Platform-specific master glossaries for the Progress documentation set. These books are in electronic format only.

Progress Master Index and Glossary for Windows and *Progress Master Index and Glossary for Character* (Hard copy only)

Platform-specific master indexes and glossaries for the Progress hard-copy documentation set.

Progress Startup Command and Parameter Reference

A reference manual that describes the Progress startup commands and parameters in alphabetical order.

Welcome to Progress (Hard copy only)

A booklet that explains how Progress software and media are packaged. An icon-based map groups the documentation by functionality, providing an overall view of the documentation set. *Welcome to Progress* also provides descriptions of the various services Progress Software Corporation offers.

Development Tools

Progress ADM 2 Guide

A guide to using the Application Development Model, Version 2 (ADM 2) application architecture to develop Progress applications. It includes instructions for building and using Progress SmartObjects.

Progress ADM 2 Reference

A reference for the Application Development Model, Version 2 (ADM 2) application. It includes descriptions of ADM 2 functions and procedures.

Progress AppBuilder Developer's Guide (Windows only)

A programmer's guide to using the Progress AppBuilder visual layout editor. AppBuilder is a Rapid Application Development (RAD) tool that can significantly reduce the time and effort required to create Progress applications.

Progress Basic Development Tools (Character only; information for Windows is in online help)

A guide for the Progress development toolset, including the Progress Procedure Editor and the Application Compiler.

Progress Debugger Guide

A guide for the Progress Application Debugger. The Debugger helps you trace and correct programming errors by allowing you to monitor and modify procedure execution as it happens.

Progress Help Development Guide (Windows only)

A guide that describes how to develop and integrate an online help system for a Progress application.

Progress Translation Manager Guide (Windows only)

A guide that describes how to use the Progress Translation Manager tool to manage the entire process of translating the text phrases in Progress applications.

Progress Visual Translator Guide (Windows only)

A guide that describes how to use the Progress Visual Translator tool to translate text phrases from procedures into one or more spoken languages.

Reporting Tools

Progress Report Builder Deployment Guide (Windows only)

An administration and development guide for generating Report Builder reports using the Progress Report Engine.

Progress Report Builder Tutorial (Windows only)

A tutorial that provides step-by-step instructions for creating eight sample Report Builder reports.

Progress Report Builder User's Guide (Windows only)

A guide for generating reports with the Progress Report Builder.

Progress Results Administration and Development Guide (Windows only)

A guide for system administrators that describes how to set up and maintain the Results product in a graphical environment. This guide also describes how to program, customize, and package Results with your own products. In addition, it describes how to convert character-based Results applications to graphical Results applications.

Progress Results User's Guide for Windows and *Progress Results User's Guide for UNIX*

Platform-specific guides for users with little or no programming experience that explain how to query, report, and update information with Results. Each guide also helps advanced users and application developers customize and integrate Results into their own applications.

4GL

Building Distributed Applications Using the Progress AppServer

A guide that provides comprehensive information about building and implementing distributed applications using the Progress AppServer. Topics include basic product information and terminology, design options and issues, setup and maintenance considerations, 4GL programming details, and remote debugging.

Progress External Program Interfaces

A guide to accessing non-Progress applications from Progress. This guide describes how to use system clipboards, UNIX named pipes, Windows dynamic link libraries, Windows dynamic data exchange, Windows ActiveX controls, and the Progress Host Language Call Interface to communicate with non-Progress applications and extend Progress functionality.

Progress Internationalization Guide

A guide to developing Progress applications for markets worldwide. The guide covers both internationalization—writing an application so that it adapts readily to different locales (languages, cultures, or regions)—and localization—adapting an application to different locales.

Progress Language Reference

A three-volume reference set that contains extensive descriptions and examples for each statement, phrase, function, operator, widget, attribute, method, and event in the Progress language.

Progress Programming Handbook

A two-volume handbook that details advanced Progress programming techniques.

Database

Progress Database Design Guide

A guide that uses a sample database and the Progress Data Dictionary to illustrate the fundamental principles of relational database design. Topics include relationships, normalization, indexing, and database triggers.

Progress Database Administration Guide and Reference

This guide describes Progress database administration concepts and procedures. The procedures allow you to create and maintain your Progress databases and manage their performance.

DataServers

Progress DataServer Guides

These guides describe how to use the DataServers to access non-Progress databases. They provide instructions for building the DataServer modules, a discussion of programming considerations, and a tutorial. Each DataServer has its own guide, for example, the *Progress DataServer for ODBC Guide*, the *Progress DataServer for ORACLE Guide*, or the *Progress/400 Product Guide*.

MERANT ODBC Branded Driver Reference

The Enterprise DataServer for ODBC includes MERANT ODBC drivers for all the supported data sources. For configuration information, see the MERANT documentation, which is available as a PDF file in *installation-path*\odbc. To read this file you must have the Adobe Acrobat Reader Version 3.1 or higher installed on your system. If you do not have the Adobe Acrobat Reader, you can download it from the Adobe Web site at: http://www.adobe.com/prodindex/acrobat/readstep.html.

SQL-89/Open Access

Progress Embedded SQL-89 Guide and Reference

A guide to Progress Embedded SQL-89 for C, including step-by-step instructions on building ESQL-89 applications and reference information on all Embedded SQL-89 Preprocessor statements and supporting function calls. This guide also describes the relationship between ESQL-89 and the ANSI standards upon which it is based.

Progress Open Client Developer's Guide

A guide that describes how to write and deploy Java and ActiveX applications that run as clients of the Progress AppServer. The guide includes information about how to expose the AppServer as a set of Java classes or as an ActiveX server.

Progress SQL-89 Guide and Reference

A user guide and reference for programmers who use interactive Progress/SQL-89. It includes information on all supported SQL-89 statements, SQL-89 Data Manipulation Language components, SQL-89 Data Definition Language components, and supported Progress functions.

SQL-92

Progress Embedded SQL-92 Guide and Reference

A guide to Progress Embedded SQL-92 for C, including step-by-step instructions for building ESQL-92 applications and reference information about all Embedded SQL-92 Preprocessor statements and supporting function calls. This guide also describes the relationship between ESQL-92 and the ANSI standards upon which it is based.

Progress JDBC Driver Guide

A guide to the Java Database Connectivity (JDBC) interface and the Progress SQL-92 JDBC driver. It describes how to set up and use the driver and details the driver's support for the JDBC interface.

Progress ODBC Driver Guide

A guide to the ODBC interface and the Progress SQL-92 ODBC driver. It describes how to set up and use the driver and details the driver's support for the ODBC interface.

Progress SQL-92 Guide and Reference

A user guide and reference for programmers who use Progress SQL-92. It includes information on all supported SQL-92 statements, SQL-92 Data Manipulation Language components, SQL-92 Data Definition Language components, and Progress functions. The guide describes how to use the Progress SQL-92 Java classes and how to create and use Java stored procedures and triggers.

Deployment

Progress Client Deployment Guide

A guide that describes the client deployment process and application administration concepts and procedures.

Progress Developer's Toolkit

A guide to using the Developer's Toolkit. This guide describes the advantages and disadvantages of different strategies for deploying Progress applications and explains how you can use the Toolkit to deploy applications with your selected strategy.

Progress Portability Guide

A guide that explains how to use the Progress toolset to build applications that are portable across all supported operating systems, user interfaces, and databases, following the Progress programming model.

WebSpeed

Getting Started with WebSpeed

Provides an introduction to the WebSpeed Workshop tools for creating Web applications. It introduces you to all the components of the WebSpeed Workshop and takes you through the process of creating your own Intranet application.

WebSpeed Installation and Configuration Guide

Provides instructions for installing WebSpeed on Windows and UNIX systems. It also discusses designing WebSpeed environments, configuring WebSpeed Brokers, WebSpeed Agents, and the NameServer, and connecting to a variety of data sources.

WebSpeed Developer's Guide

Provides a complete overview of WebSpeed and the guidance necessary to develop and deploy WebSpeed applications on the Web.

WebSpeed Product Update Bulletin

A booklet that provides a brief description of each new feature of the release. The booklet also explains where to find more detailed information in the documentation set about each new feature.

Welcome to WebSpeed! (Hard copy only)

A booklet that explains how WebSpeed software and media are packaged. *Welcome to WebSpeed!* also provides descriptions of the various services Progress Software Corporation offers.

Reference

Pocket Progress (Hard copy only)

A reference that lets you quickly look up information about the Progress language or programming environment.

Pocket WebSpeed (Hard copy only)

A reference that lets you quickly look up information about the SpeedScript language or the WebSpeed programming environment.

1

Database Tasks

This chapter describes how to start the Data Dictionary. It also describes the database tasks you can perform using the Data Dictionary:

- Creating a database
- Connecting a database
- Switching to another connected database
- Disconnecting a database
- Viewing session information
- Dumping data definitions
- Creating an incremental data definition file
- Creating a parameter file
- Editing a parameter file
- Loading data definitions

1.1 Starting the Data Dictionary

There are three ways to start the Data Dictionary:

- At the command line, enter the following command:

```
pro -p dict.p
```

- From any of the other ADE tools, choose Tools→ Data Dictionary.
- In the Procedure Editor, type **dict**. Then choose Compile→ Run or press **GO**.

When you start the Data Dictionary, Progress displays the window shown in Figure 1–1.

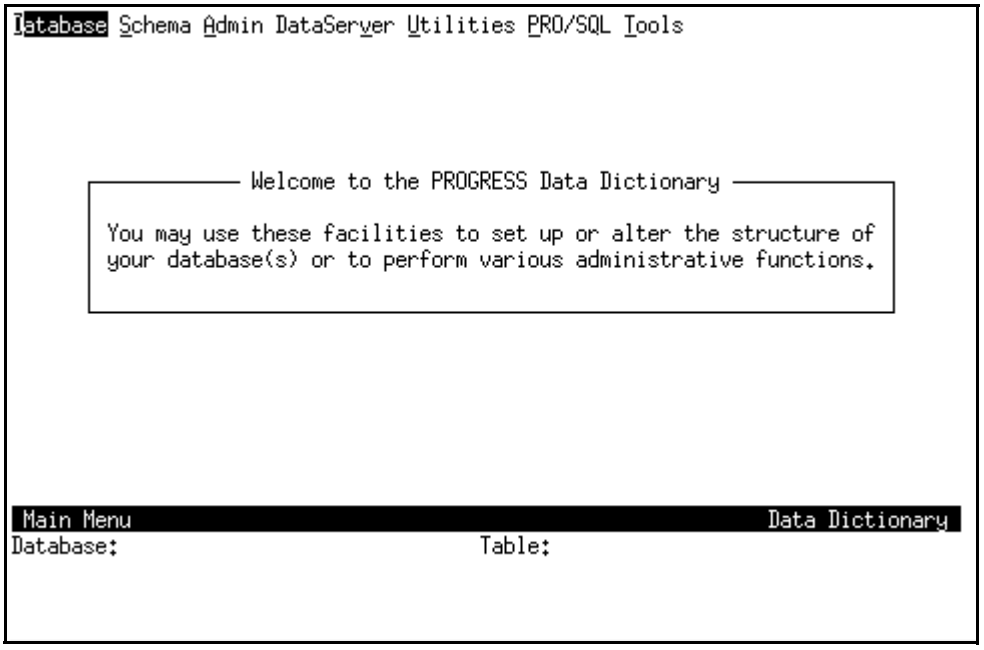


Figure 1–1: Data Dictionary Window

If you have at least one connected database, Progress displays the name of the working database in the lower-left corner of the window. You can have multiple databases connected at any time, but only one is the current, *working database*. To select an existing working database, choose Database→ Select Working Database from the menu bar.

1.2 Database Tasks

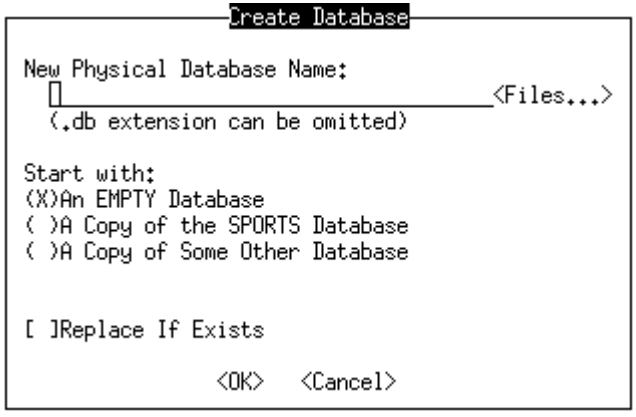
This section describes the following database tasks you can perform with the Data Dictionary:

- Creating a database
- Connecting a database
- Switching to another connected database
- Disconnecting a database
- Viewing session information
- Dumping data definitions
- Creating an incremental data definition file
- Creating a parameter file
- Editing a parameter file
- Loading data definitions

1.2.1 Creating a Database

Follow these steps to create a database:

1 ♦ Choose Database→ Create. Progress displays the Create Database dialog box:



2 ♦ Enter the name of the new database.

3 ♦ Specify whether you want to create an empty database, a copy of the sports database, or a copy of an existing Progress database.

4 ♦ Specify whether you want to overwrite any existing database with the same name, then choose OK.

If you do not choose the Replace If Exists button, and a database already exists with the same name, Progress prompts you to verify that you want to overwrite it.

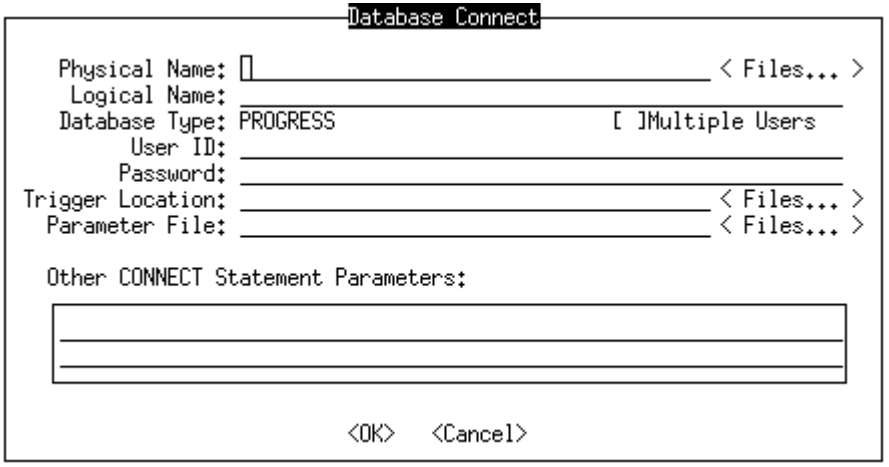
If no database with that name exists, Progress creates the database, or if one does and you chose the Replace If Exists button, Progress overwrites the existing database and the Database Connect dialog box appears. Define the database attributes, then choose OK. See Chapter 8, "Database Menu Reference," for a complete description of the database attributes you can specify in the Database Connect dialog box.

If more than one database is connected, the Select Working Database dialog box appears. After you select a working database, Progress returns you to the Data Dictionary window.

1.2.2 Connecting a Database

Follow these steps to connect a database:

1 ♦ Choose Database→ Connect. Progress displays the Database Connect dialog box:



2 ♦ Specify the database name and any necessary database information, then choose OK. See Chapter 8, "Database Menu Reference," for a complete description of the database attributes you can specify in the Database Connect dialog box.

NOTE: You **must** select the Multiple Users toggle-box if you are running a multi-server and trying to connect to a self-service client.

1.2.3 Switching to Another Connected Database

Follow these steps to switch to another connected database:

1 ♦ Choose Database→ Select Working Database. Progress displays a list of the connected databases.

2 ♦ Highlight the database you want for the working database, then press **RETURN**.

1.2.4 Disconnecting a Database

Follow these steps to disconnect a database:

1 ♦ Choose Database→ Disconnect. If you have more than one database connected, Progress displays the Select Database to Disconnect dialog box.

2 ♦ Select the database you want to disconnect from the database selection list, then press **RETURN**. Progress displays an alert box prompting you to verify the disconnect.

3 ♦ Choose Yes to disconnect the selected database.

If you have more than one database connected, the Select Working Database dialog box appears and lets you select another database as the working database.

NOTE: If you disconnect a schema holder, Progress also disconnects all of the schema holder's corresponding non-Progress databases.

1.2.5 Viewing Session Information

To display current database session information, choose Utilities→ Information. Progress displays the Session Information dialog box, which contains information such as how many databases are connected, the physical and logical names of the working database, and the database type and version. See Chapter 11, "Advanced Menus Reference," for more information on the Session Information dialog box.

1.2.6 Dumping Data Definitions

Follow these steps to dump data definitions:

1 ♦ Choose Admin→ Dump Data and Definitions→ Data Definitions (.df file). Progress alphabetically lists all the tables defined for your database.

2 ♦ Select the table for which you want to dump the definitions or type **ALL**.

If you type ALL, the .df file will also contain the sequence and auto-connect record definitions, but not the collation/conversion table. Progress displays a default name for the output file into which you can dump your data definitions (hidden tables are not dumped). This default file is the name of the dump file you specified when you created the table or, for multiple tables, the database name with the .df extension. Progress truncates file names to eight characters.

NOTE: When you dump only one table, the table's dump name becomes the default for its corresponding contents dump file. For example, if you select the Order-line table, the default filename for its corresponding contents dump file is `order-line.d`. If you specify to dump all the tables, the default name for the file contents is *`db-name`*`.d`. See Chapter 2, "Table Tasks," for information on how to dump table contents.

3 ♦ Specify the filename or use the default value.

4 ♦ Specify the code page or accept the default, which is the value specified by the SESSION:STREAM or SESSION:CHARSET attribute, then choose OK. For more information on the SESSION:STREAM or SESSION:CHARSET attributes, see the chapter on user environments in the *Progress Client Deployment Guide*.

Progress displays each object name as it writes its definition to the .df file. Progress displays a status message and prompts you to continue.

1.2.7 Creating an Incremental Data Definition File

An incremental data definition file contains the differences when you compare two versions of a Progress database schema.

Follow these steps to create an incremental data definition file:

1 ♦ Choose Database→ Select Working Database, then select the new version of the database as the working database.

2 ♦ Choose Admin→ Dump Data and Definitions→ Create Incremental .df File. Progress lists all your connected databases on the Create Incremental .df File window.

3 ♦ Select the database you want to compare the working database against (that is, the old version of the database). Progress then prompts you for the filename to which you want to write the differences.

4 ♦ Enter the filename to which you want to write the differences. The default filename is `delta.df`.

5 ♦ Specify a code page.

6 ♦ Choose OK. Progress displays the table, field, index, and sequence names as it compares the databases and prompts you to specify what you want to do with any differences it finds. Progress displays a status message when it has completed dumping the table definitions.

7 ♦ Choose OK to return to the menu bar.

1.2.8 Creating a Parameter File

Follow these steps to create a parameter file:

1 ♦ Choose Utilities→ Editor for Parameter Files. Progress prompts you for the file you want to create.

2 ♦ Specify the filename. Progress alphabetically lists all available startup parameters.

3 ♦ Specify which of the parameters you want in the file by entering yes, no, or the appropriate value, then press **GO**. Progress prompts you for comments.

4 ♦ Enter any comments you want to include in the parameter file. After creating the parameter file, Progress returns you to the Data Dictionary window.

1.2.9 Editing a Parameter File

Follow these steps to edit a parameter file:

1 ♦ Choose Utilities→ Editor for Parameter Files. Progress prompts you for the file you want to edit.

2 ♦ Specify the name of an existing parameter file. Progress alphabetically lists all available startup parameters with the values for the specified file.

3 ♦ Modify the file by entering yes, no, or the appropriate value, then press **GO**. Progress prompts you for comments.

4 ♦ Enter any comments you want to include in the parameter file. After saving the parameter file, Progress returns you to the Data Dictionary window.

1.2.10 Loading Data Definitions

Follow these steps to load data definitions:

1 ♦ Choose Admin→ Load Data and Definitions→ Data Definitions (.df file). Progress prompts you for the name of the file that contains the data definitions you want to load into your current working database.

2 ♦ Specify a filename or accept the default. The default filename is the working physical database name with a .df extension.

3 ♦ Specify whether you want to stop the load as soon as Progress encounters a bad definition statement, then choose OK.

If the data definitions file does not specify a code page, Progress prompts you for one.

Progress displays each item as it loads the definitions for that object. Progress displays a status message and prompts you to continue.

2

Table Tasks

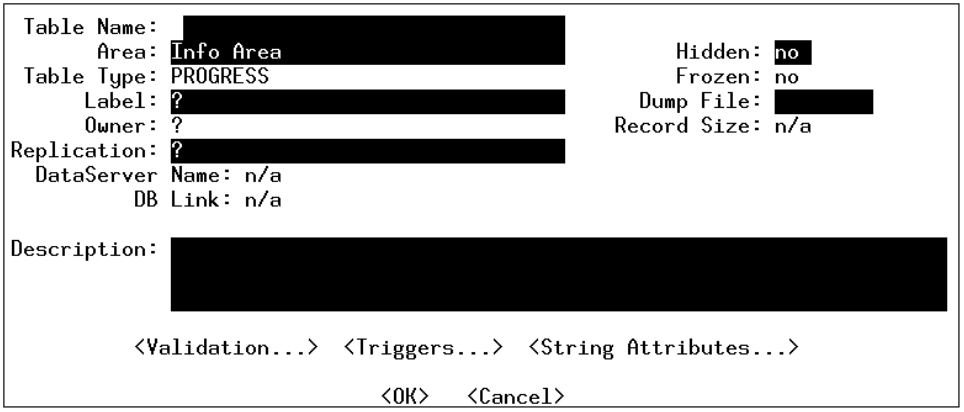
This chapter describes how to perform the following administrative tasks for tables and files:

- Creating a table
- Modifying a table
- Deleting a table
- Freezing/unfreezing a table
- Generating a table report
- Dumping table contents
- Loading table contents
- Binary dump and load
- Creating a bulk loader description file
- Generating an include file

2.1 Creating a Table

Follow these steps to create a table:

1 ♦ Choose Schema→ Add New Table. Progress displays the following window:

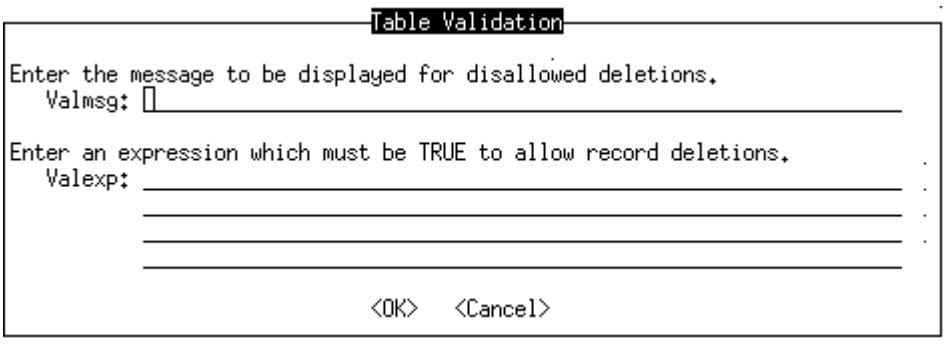


2 ♦ Define the table attributes. For more information on the New Table window, see Chapter 9, "Schema Menu Reference."

2.1.1 Specifying Validation Criteria

Follow these steps to specify table validation criteria:

1 ♦ Choose the Validation button. Progress displays the Table Validation dialog box:



2 ♦ Specify the message you want Progress to display when a user tries an invalid deletion. There is no limit to the number of characters you can enter. A message up to 63 characters long can fit onto the two message lines without scrolling.

3 ♦ Specify the criteria for valid deletions. For example, in the customer table, to restrict deleting customers to those customers who have no outstanding orders, enter the following criteria:

```
NOT (CAN-FIND(FIRST order OF customer))
```

4 ♦ Choose OK to return to the Add New Table window.

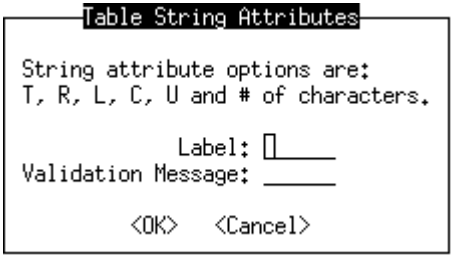
See the "Creating a Trigger" section in Chapter 6, "Trigger Tasks," for more information on creating triggers.

2.1.2 Specifying String Attributes

Follow these steps to specify table string attributes:

NOTE: For more information on string attributes, see the "SchemaÆ Modify Table" section in Chapter 9, "Schema Menu Reference."

1 ♦ Choose the String Attributes button. Progress displays the Table String Attributes dialog box:



2 ♦ Specify the attribute characters for the table label and the table validation message.

3 ♦ Choose OK to return to the Create Table window.

2.2 Modifying a Table

When you modify a field, sequence, trigger, or index structure, you modify the table. Follow these steps to modify a table definition:

1 ♦ Choose Schema→ Modify Table. Progress alphabetically lists all the tables defined for your database.

2 ♦ Select the table you want to modify. Progress displays the Modify Table window.

3 ♦ Modify the table definition.

NOTE: See the earlier "Creating a Table" section for information on specifying table validation and string attributes. See Chapter 6, "Trigger Tasks," for information about modifying triggers.

4 ♦ Choose OK. A list of options appears.

5 ♦ Choose A to apply the changes to your database.

2.3 Deleting a Table

Follow these steps to delete a table:

1 ♦ Choose Schema→ Delete Table(s) from the menu bar. Progress alphabetically lists all the tables defined for your database.

2 ♦ Select the table you want to delete or type ALL to delete all tables, then press **GO**. Progress prompts you to verify that you want to delete the selected table.

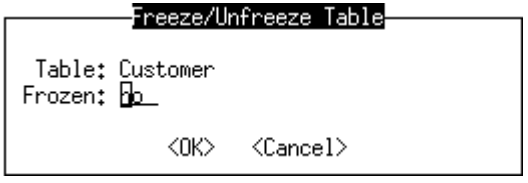
3 ♦ Choose Yes. Progress deletes the table from the database.

2.4 Freezing/Unfreezing a Table

Follow these steps to freeze or unfreeze a table:

1 ♦ Choose Utilities→ Freeze/Unfreeze. Progress alphabetically lists all the tables defined for your database.

2 ♦ Select the table you want to freeze or unfreeze. Progress displays the Freeze/Unfreeze Table dialog box:



3 ♦ Specify yes or no in the Frozen fill-in field.

4 ♦ Choose OK to return to the Data Dictionary window.

2.5 Generating a Table Report

This section describes how to generate the following table reports:

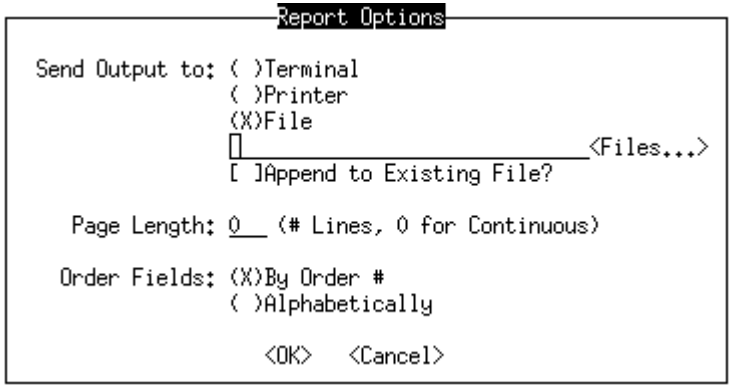
- A detailed table report
- A quick table report
- A table relations report

2.5.1 Generating a Detailed Table Report

Follow these steps to generate a detailed table report:

1 ♦ Choose Database→ Reports→ Detailed Table. Progress alphabetically lists all the tables defined for your database.

2 ♦ Select a table or type **ALL**. The Report Options dialog box appears:



3 ♦ Specify the report destination. You can display the report on your terminal, print it, or save it to a file. If you want to save the report to a file, specify the filename and whether to append the report to an existing file.

4 ♦ Specify the page length.

5 ♦ Specify to order the fields by order number or alphabetically, then choose OK.

If you specified a file or printer, Progress generates the report to the specified output destination and returns you to the Data Dictionary window.

If you sent the report to the terminal, the Detailed Table Report dialog box appears. You can choose the Change Field Order button to reorder the fields, choose the Print button to print the report or write it to a file, or choose the OK button to close the Report Options dialog box.

2.5.2 Generating a Quick Table Report

Follow these steps to generate a quick table report:

1 ♦ Choose Database→ Reports→ Quick Table. The Quick Table Report dialog box appears and displays information about all the tables in the working database.

2 ♦ Choose the Print button if you want to print the report or to write it to a file. The Print Options dialog box appears.

3 ♦ Specify the report destination. If you want to save the report to a file, specify the filename and whether to append the report to an existing file.

4 ♦ Specify the page length and choose OK. Progress sends the report to the specified output destination and returns you to the Quick Table Report dialog box.

5 ♦ Choose OK to return to the Data Dictionary window.

2.5.3 Generating a Table Relations Report

Follow these steps to generate a table relations report:

1 ♦ Choose Database→ Reports→ Table Relations. Progress alphabetically lists all the tables defined for your database.

2 ♦ Select a table or type **ALL**. Progress generates the report.

3 ♦ Choose the Print button if you want to print the report or write it to a file. The Print Options dialog box appears.

4 ♦ Specify the report destination. If you want to save the report to a file, specify the filename and whether to append the report to an existing file.

5 ♦ Specify the page length and choose OK. Progress sends the report to the specified output destination and returns you to the Table Relations Report dialog box.

6 ♦ Choose Switch Tables to generate a table relations report for another table or choose OK to return to the Data Dictionary window.

2.6 Dumping Table Contents

When you dump table data, Progress creates a contents file that contains data for each table you specify. This file has a .d extension. Progress uses the contents file to load data into your database. Progress creates a separate contents file for every table you dump.

For information about performing a binary dump and load, see the "Binary Dump and Load" section later in this chapter.

To dump table contents, follow these steps:

1 ♦ Access the Data Dictionary tool.

2 ♦ Make sure you are using the database (the source) from which you want to dump the table data.

3 ♦ Choose Admin→ Dump Data and Definitions→ Table Contents (.d files) from the menu bar. Progress alphabetically lists all the tables defined for your database (hidden tables do **not** display).

4 ♦ Select the table names whose contents you want to dump. Use the Select Some and Deselect Some buttons to select or deselect groups of tables.

5 ♦ Press **GO**.

If you selected a single table, Progress displays a default name for the file that you can dump your table contents into. This default file name is always the dump name of the table definition file, with a .d extension. If you want to specify a file name other than the default file, type the name in the Output File field.

If you selected more than one table, Progress prompts you for a directory name to write the contents files. If you do not specify a directory, Progress creates the files in the current directory. Progress names each contents file with the table's corresponding dump name.

6 ♦ Accept the default or enter a different name.

7 ♦ If you want to use character mapping, enter the character mapping, then choose OK. See the *Progress Internationalization Guide* for information about character mapping, PROTERMCAP, and national language support.

8 ♦ Specify the code page or accept the default.

Progress displays each table name as it writes the table contents to the table contents file. After dumping the table contents, Progress displays a status message. Choose OK to return to the Data Dictionary window.

9 ♦ Choose OK to dump the data contents file(s) and return to the Data Dictionary main window.

NOTE: You must dump your sequence values separately.

2.6.1 Dumping User Table Contents

Follow these steps to dump the user table contents:

1 ♦ Access the Data Dictionary tool.

2 ♦ Make sure you are using the database from which you want to dump the table data.

3 ♦ Choose Admin→ Dump Data and Definitions→ User Table Contents. Progress prompts you for the file to which to write the user file contents. The default filename is `_user.d`.

4 ♦ Specify the filename or accept the default. After dumping the user file contents to the specified file, Progress displays a status message and prompts you to continue.

5 ♦ Choose OK to return to the Data Dictionary main window.

2.7 Loading Table Contents

Progress uses a contents file that you specify to load table data into your database. A contents file contains all the data for a table.

For information about performing a binary dump and load, see the "Binary Dump and Load" section later in this chapter.

NOTES

- Table definitions must be in the database before you can load table contents.
- You must load your sequence values separately.

There are two ways to load the table contents:

- Use the Data Dictionary tool.
- Use the Bulk Loader (PROUTIL BULKLOAD) utility. For information on using the Bulk Loader, see the chapter on administration utilities in the *Progress Database Administration Guide and Reference*.

To load table contents into a database, follow these steps:

1 ♦ Access the Data Dictionary tool.

2 ♦ Make sure that the working database is the target database where you want to load the table contents.

3 ♦ Choose Admin→ Load Data and Definitions→ Table Contents (.d files) from the menu bar. Progress alphabetically lists all the tables defined for your database.

4 ♦ Select the table names whose contents you want to load. Use the Select Some and Deselect Some buttons to select or deselect groups of tables.

Progress prompts you for the name of the contents file—or if you choose multiple files, the directory that contains the contents files—you want to load into your current database.

5 ♦ Specify the directory name, filename, or use the default value.

When you load the contents of more than one table, enter a directory from which you want to load the contents files. If you do not specify a directory, Progress loads the files from the current directory. Progress loads each table from the corresponding *table-dumpname*.d filename.

Progress also prompts you for an acceptable error rate.

6 ♦ Specify the error rate. As Progress loads records from any files you designate, it might encounter data that cannot be loaded. For example, a record might have data in one of its fields that is too long for that field's defined format. Progress does not load the record. It places all erroneous records in a *table*.e file (the name of the table with a .e extension). If you specify an error rate of 10 percent, Progress must successfully load 90 records from every set of 100 records loaded.

7 ♦ Choose OK. After loading the table contents, Progress displays a status message and prompts you to continue.

8 ♦ Choose OK to return to the Data Dictionary main window.

2.7.1 Loading User Table Contents

Follow these steps to load the user table contents:

1 ♦ Access the Data Dictionary tool.

2 ♦ Choose Admin→ Load Data and Definitions→ User Table Contents. Progress prompts you for the file from which you want to read the user file contents. The default filename is _user.d.

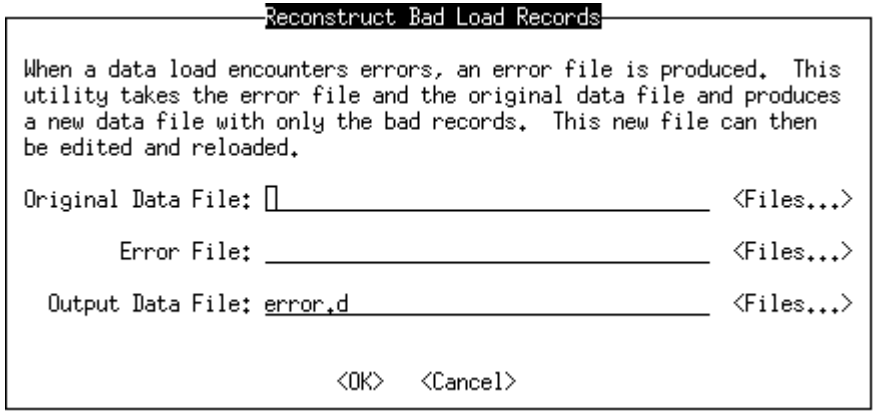
3 ♦ Specify the filename or accept the default. After loading the user file contents to the specified file, Progress displays a status message and prompts you to continue.

4 ♦ Choose OK to return to the Data Dictionary main window.

2.7.2 Reconstructing Bad Load Records

If Progress encounters errors while loading a data file, it creates an error file. Follow these steps to use this error file and original data file to build a new data file for bad records:

1 ♦ Choose Admin→ Load Data and Definitions→ Reconstruct Bad Load Records. The following dialog box appears:



2 ♦ Specify the filename of the original data file, the error file, and the new data file. The default filename for the new data file is `error.d`.

3 ♦ Choose OK.

Progress displays a status message and prompts you to continue.

2.8 Binary Dump and Load

Progress allows you to move and reorganize data by performing a binary dump and load instead of an ASCII dump and load. The binary method improves the dump and load performance significantly because the data remains in binary format during the process.

Performing the Binary Dump

Use the following PROUTIL command to perform a binary dump:

```
proutil dbname -C dump tablename directory [ -index num ]
```

db-name

Specifies the database from which you want to dump. You must completely define the path.

tablename

Specifies the name of the table containing the data you want to dump.

directory

Specifies the name of the target directory into which the data will be dumped.

-index num

Specifies an index with which to dump the table's contents. If you choose not to use this option, the command uses the primary index to dump the table.

The dump command writes the data to a dump file named table.bd, where *table* is the name of the table that was dumped. For example, if you dump data from the table named "customer," the procedure creates a dump file named customer.bd in the directory you specify in the command.

Each binary dump file contains a header and a description for each record in the table. The dump file appears in the following format:

Header	Record length	Table number	Binary record	Record CRC

The file header contains information that appears in this order:

- Version number
- Date and time the file was created
- Name of the table being dumped
- Number of the table being dumped
- CRC of the table being dumped
- Number of fields in the table
- Name of the database in which the table resides
- Section number
- Number of the first record

When the dump procedure finishes, a screen message reports the number of records that were dumped and the number of kilobytes of data that were dumped.

2.8.1 Performing the Binary Load

Use the following PROUTIL command to perform a binary load:

```
proutil dbname -C load filename
```

db-name

Specifies the database into which you want to load the data. You must completely define the path.

filename

Specifies the binary dump file that you want to load. You must completely define the path.

When the load procedure finishes, a screen message reports the number of records that were loaded.

2.9 Creating a Bulk Loader Description File

You use the bulk loader description file with the Bulk Loader utility to load data files into a Progress database more quickly than you can using the Data Dictionary Load utility.

You must create a bulk loader description file before you can run the Bulk Loader utility. For more information about bulk loader description files, see the chapter on dumping and loading in the *Progress Database Administration Guide and Reference*.

NOTE: You cannot use the Bulk Loader utility for non-Progress databases.

Follow these steps to create a bulk loader description file:

1 ♦ Choose Admin→ Create Bulk Loader Description File. Progress alphabetically lists all the tables defined for your database.

2 ♦ Select the tables for which you want to create bulk loader description files and press **GO**. Progress prompts you for the bulk load filename.

3 ♦ Specify a bulk load filename. The default filename is *table-name*.fd.

4 ♦ Specify the code page.

5 ♦ Choose OK. After dumping the table contents to the specified file, Progress displays a status message and prompts you to continue.

2.10 Generating an Include File

You can generate include files for the following statements:

- ASSIGN
- FORM
- DEFINE WORK-TABLE

2.10.1 Generating an ASSIGN Statement Include File

Follow these steps to create an ASSIGN statement include file:

1 ♦ Choose Utilities→ Generate Include Files→ ASSIGN statement. Progress alphabetically lists all the tables defined for your database.

2 ♦ Select the table for which you want to create the ASSIGN statement. Progress prompts you for the filename of the output file.

3 ♦ Specify the filename. The default filename is the *table-name*.i.

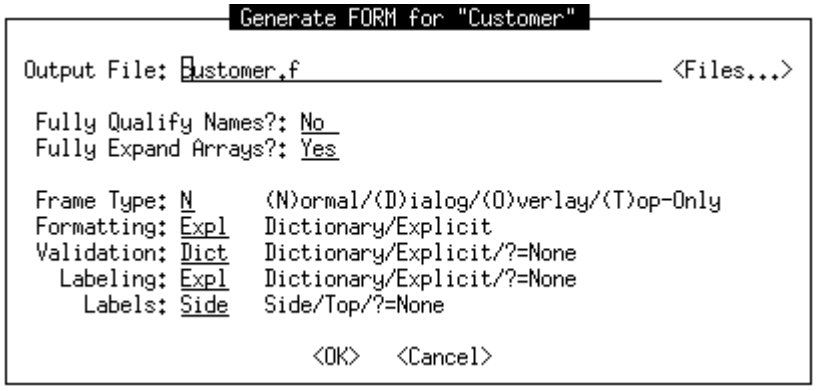
4 ♦ Choose OK. Progress displays a status message and prompts you to continue.

2.10.2 Generating a FORM Statement Include File

Follow these steps to create a FORM statement include file:

1 ♦ Choose Utilities→ Generate Include Files→ FORM statement. Progress alphabetically lists all the tables defined for your database.

2 ♦ Select the table for which you want to create the FORM statement. The Generate FORM dialog box appears:



3 ♦ Enter the appropriate information in the fill-in fields, then choose OK. Progress displays a status message and prompts you to continue.

2.10.3 Generating a DEFINE WORK-TABLE Statement Include File

Follow these steps to create a DEFINE WORK-TABLE statement include file:

1 ♦ Choose Utilities→ Generate Include Files→ DEFINE WORK-TABLE statement. Progress alphabetically lists all the tables defined for your database.

2 ♦ Select the table for which you want to create the DEFINE WORK-TABLE statement. Progress prompts you for the filename of the output file.

3 ♦ Specify the filename. The default filename is the *table-name*.i.

4 ♦ Choose OK. Progress displays a status message and prompts you to continue.

3

Field Tasks

This chapter describes the following tasks involved in creating and maintaining fields:

- Adding a field
- Modifying a field
- Changing a field name globally
- Copying a field
- Deleting a field
- Changing field order
- Generating a quick field report
- Adjusting the SQL field width

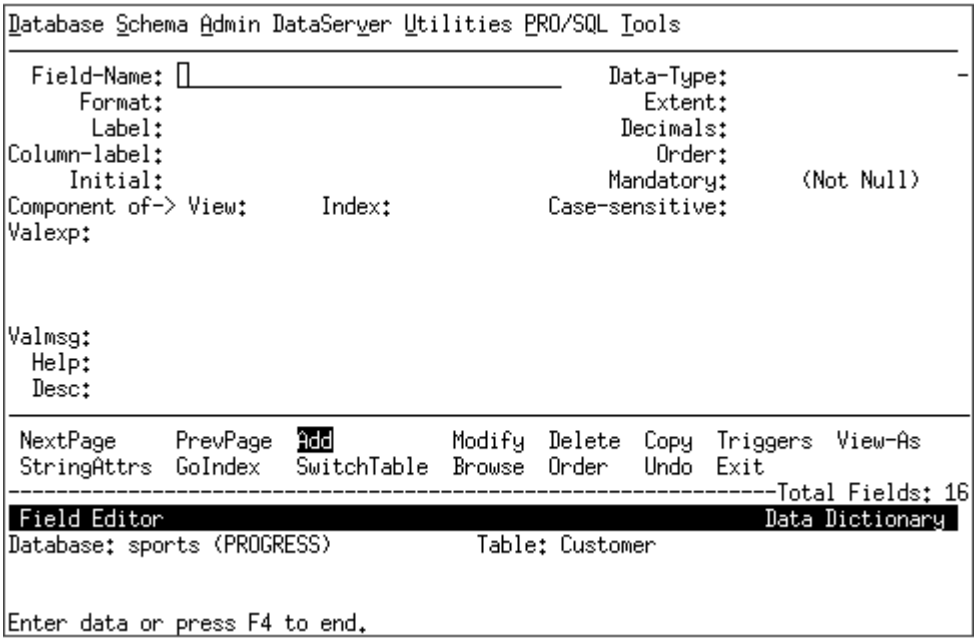
3.1 Adding a Field

Follow these steps to add a field to a table:

1 ♦ Choose Schema→ Field Editor. Progress alphabetically lists all the tables defined for your database.

2 ♦ Select the table to which you want to add a new field. Progress lists all the fields defined for the selected table.

3 ♦ Choose the Add option from the horizontal menu at the bottom of the window. A window similar to the one below appears:



4 ♦ Define the field attributes. See the Add Option subsection of the "SchemaÆ Field Editor" section in Chapter 9, "Schema Menu Reference," for more information about the fields in this window.

5 ♦ Press **RETURN** or **GO**.

6 ♦ Choose Exit to apply the change. Progress displays an alert box asking if you want to apply or undo the changes to the database, or if you want to make other changes.

7 ♦ Choose Apply Changes. Progress returns you to the Data Dictionary main window.

3.2 Modifying a Field

Follow these steps to modify a field definition:

1 ♦ Choose Schema→ Field Editor. Progress alphabetically lists all the tables defined for your database.

2 ♦ Select the table that contains the field you want to modify. Progress lists all the fields defined for the selected table.

3 ♦ Choose the Modify option from the horizontal menu at the bottom of the window.

4 ♦ Choose the field you want to modify. Progress displays the field definition in the window.

See the Modify Option subsection in the "SchemaÆ Field Editor" section of Chapter 9, "Schema Menu Reference," for information about the fields on this window.

5 ♦ Edit the field definition.

6 ♦ Press **RETURN** or **GO**.

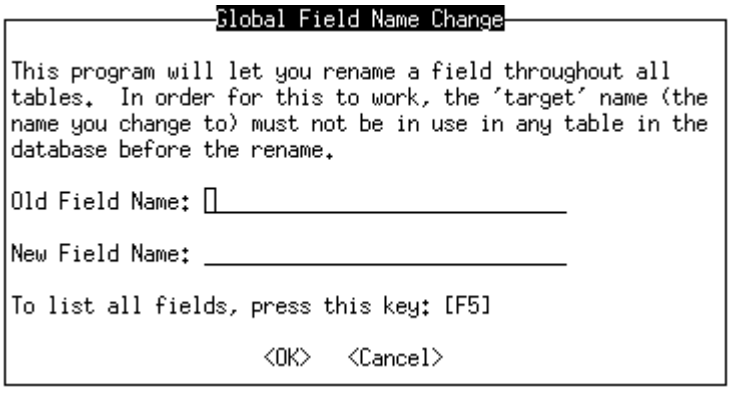
7 ♦ Choose Exit to apply the change. Progress displays an alert box asking if you want to apply or undo the changes to the database, or if you want to make other changes.

8 ♦ Choose Apply Changes. Progress returns you to the Data Dictionary main window.

3.3 Changing a Field Name Globally

Follow these steps to globally change a field name:

1 ♦ Choose Schema→ Global Field Name Change. The Global Field Name Change dialog box appears:



2 ♦ Specify the current field name. Press **F5** to display a list of all field names.

3 ♦ Specify the new name.

4 ♦ Choose OK. Progress displays an alert box prompting for verification of the global field name change.

5 ♦ Choose Yes to apply the change. Progress returns you to the Data Dictionary main window.

3.4 Copying a Field

Follow these steps to copy a field from one table to another table in the working database:

1 ♦ Choose Schema→ Field Editor. Progress alphabetically lists all the tables defined for your database.

2 ♦ Select the table to which you want to copy the field. Progress lists all the fields defined for the selected table.

3 ♦ Choose the Copy option from the menu at the bottom of the window.

4 ♦ Select the table from which you want to copy the field. Progress alphabetically lists all the fields defined for that table.

5 ♦ Select the fields you want to copy, then press **GO**. Progress copies the selected fields into the table you selected in Step 2 and returns you to the Field Editor.

6 ♦ Choose Exit to apply the change. Progress displays an alert box asking if you want to apply or undo the changes to the database, or if you want to make other changes.

7 ♦ Choose Apply Changes. Progress returns you to the Data Dictionary main window.

3.5 Deleting a Field

Follow these steps to delete a field:

1 ♦ Choose Schema→ Field Editor. Progress alphabetically lists all the tables defined for your database.

2 ♦ Select the table that contains the field you want to delete. Progress lists all the fields defined for the selected table.

3 ♦ Choose the Delete option from the menu at the bottom of the window.

4 ♦ Choose the field you want to delete, then press **RETURN**. Progress prompts you to verify that you want to delete the field.

5 ♦ Choose Yes. Progress returns you to the Field Editor.

6 ♦ Choose Exit to apply the change. Progress displays an alert box asking if you want to apply or undo the changes to the database, or if you want to make other changes.

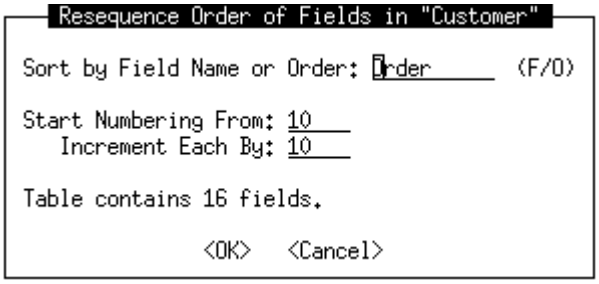
7 ♦ Choose Apply Changes. Progress returns you to the Data Dictionary main window.

3.6 Changing Field Order

Follow these steps to change field order:

1 ♦ Choose Schema→ Reorder Fields. Progress alphabetically lists all the tables defined for your database.

2 ♦ Select the table for which you want to reorder the fields. A dialog box similar to the one below appears:



3 ♦ Specify whether Progress should sort by field name or by order number.

4 ♦ Specify from where you want to start numbering and the number by which to increment.

5 ♦ Choose OK. Progress arranges the fields for the selected table in the specified order.

6 ♦ Choose OK to return to the Data Dictionary main window.

3.7 Generating a Quick Field Report

Follow these steps to generate a quick field report:

1 ♦ Choose Database→ Reports→ Quick Field. Progress alphabetically lists all the tables defined for your database.

2 ♦ Select a table or type **ALL**. Progress generates the report. The Quick Field Report dialog box appears, which lists the fields in the table by order number.

3 ♦ Choose one of the following options:

a) Choose Switch Tables to generate a quick field report for another table. Progress alphabetically lists all the tables for you to choose another table.

b) Change Field Order to reorder the fields alphabetically.

c) Choose Print to print the report or write it to a file. The Print Options dialog box appears:

- Specify the report destination. If you want to save the report to a file, specify the filename and whether to append the report to an existing file.
- Specify the page length, then choose OK. Progress sends the report to the specified output destination and returns you to the Quick Field Report dialog box.

4 ♦ Choose OK to return to the Data Dictionary window.

3.8 Adjusting the SQL Field Width

The SQL Field Width property allows the 4GL client to modify the field width so that the SQL-92 client will have the proper size to work with. If you do not adjust the setting, the SQL-92 client uses the default width property.

To adjust the field width in character mode, make sure you are connected to a database and follow these steps:

1 ♦ Access the Data Dictionary.

2 ♦ Choose Schema→ SQL Properties. A table name appears in the Table Name field, which is followed by a list of additional table names.

3 ♦ Choose the table you want; the name automatically appears in the Table Name field.

4 ♦ Press **ENTER** to open the table. The width of the first field in the field name listing is highlighted.

5 ♦ Select the name of the field whose width you want to adjust, click its width setting, and make the adjustment.

6 ♦ Repeat Step 5 for any other fields.

7 ♦ Choose Save, then press **ENTER**.

8 ♦ Press **F4**, then choose Apply Changes.

4

Index Tasks

This chapter describes the following tasks involved in creating and maintaining indexes:

- Creating an index
- Modifying an index
- Changing the primary index
- Renaming an index
- Deactivating an index
- Deleting an index
- Generating a quick index report

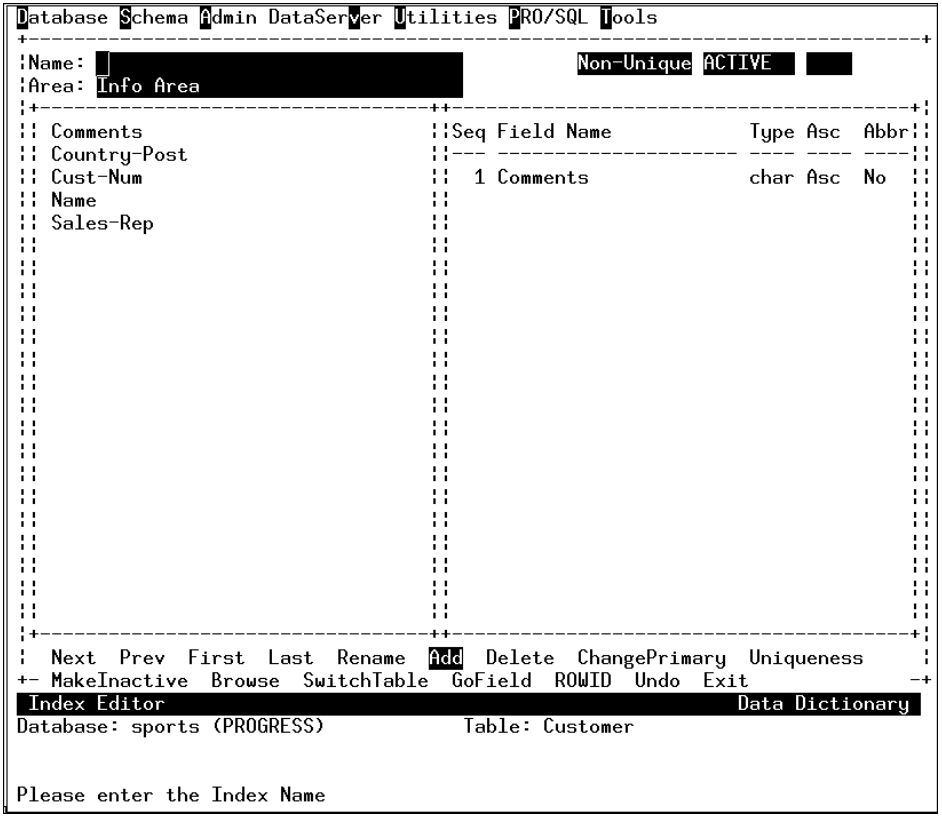
4.1 Creating an Index

Follow these steps to create an index:

1 ♦ Choose Schema→ Index Editor. Progress alphabetically lists all the tables defined for your database.

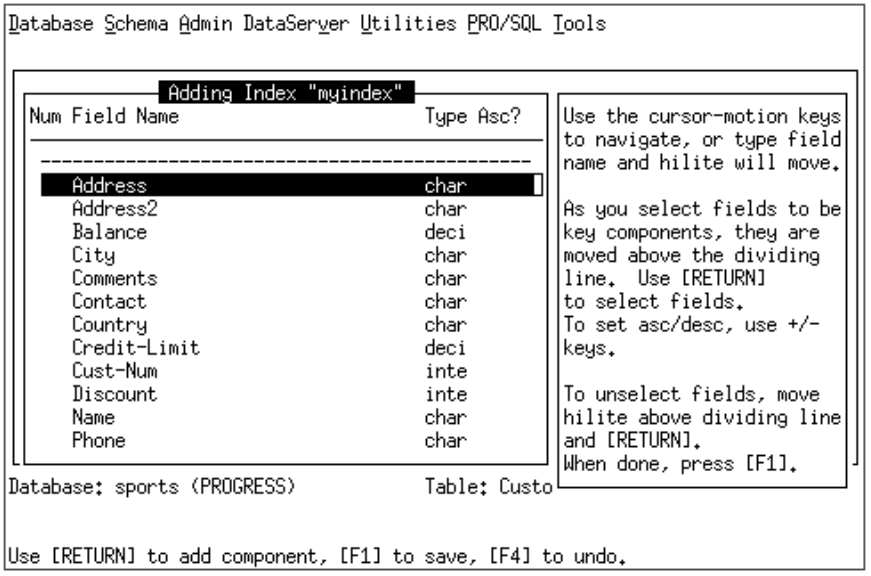
2 ♦ Select a table.

3 ♦ Choose the Add option from the bottom of the window:



4 ♦ Enter a name for the index. Specify whether the index is unique or non-unique, active or inactive, and a word index or a standard index.

5 ♦ Press **RETURN**. Progress displays the Adding Index window, which lists the field names you can add to the new index:



6 ♦ Use the up and down arrow keys to move the highlight bar over the field names. To select a highlighted field, press **RETURN**. Progress prompts you to specify whether you want this index component to be ascending or descending.

7 ♦ Specify Yes for ascending, No for descending. Progress moves the selected filed above the dividing line of the window. To change the sort order of a field above the dividing bar, highlight the field then press **+** for ascending and **-** for descending.

8 ♦ Select the next field. You can specify up to 10 fields to define an index.

9 ♦ When you are done choosing fields, press **GO**. If the index has a character field as the last index component, Progress prompts you to specify whether you want to use the Abbreviate option.

10 ♦ Specify whether to use the Abbreviate option. Progress returns you to the Index Editor and adds the new index to the list.

11 ♦ Choose Exit to apply the change. Progress displays an alert box asking if you want to apply or undo the changes to the database, or if you want to make other changes.

12 ♦ Choose Apply Changes. Progress returns you to the Data Dictionary main window.

4.2 Modifying an Index

You can use the Data Dictionary to make an index the primary index for the table, to rename an index, or to deactivate an index.

NOTE: To change the component fields of an index, such as ascending or descending order or uniqueness, you must delete and then re-create the index.

4.3 Changing the Primary Index for a Table

Follow these steps to change the primary index for a table:

NOTE: A word index cannot be the primary index.

1 ♦ Choose Schema→ Index Editor. Progress alphabetically lists all the tables defined for your database.

2 ♦ Select a table.

3 ♦ Choose the ChangePrimary option from the menu at the bottom of the window. Progress prompts you to verify that you want to make the selected index the primary index for the table.

4 ♦ Choose Yes to make the selected index the primary index for the table. Progress returns you to the Index Editor.

5 ♦ Choose Exit to apply the change. Progress displays an alert box asking if you want to apply or undo the changes to the database, or if you want to make other changes.

6 ♦ Choose Apply Changes. Progress returns you to the Data Dictionary main window.

4.4 Renaming an Index

Follow these steps to rename an index:

1 ♦ Choose Schema→ Index Editor. Progress alphabetically lists all the tables defined for your database.

2 ♦ Select a table. Progress lists the indexes defined for that table.

3 ♦ Select the index you want to rename.

4 ♦ Choose the Rename option from the menu at the bottom of the window. Progress prompts you for the new name.

5 ♦ Enter the new index name, then press **RETURN**. You cannot use the name of an existing index. Progress returns you to the Index Editor.

6 ♦ Choose Exit to apply the change. Progress displays an alert box asking if you want to apply or undo the changes to the database, or if you want to make other changes.

7 ♦ Choose Apply Changes. Progress returns you to the Data Dictionary main window.

4.5 Deactivating an Index

Follow these steps to deactivate an index:

1 ♦ Choose Schema→ Index Editor. Progress alphabetically lists all the tables defined for your database.

2 ♦ Select a table. Progress lists the indexes defined for that table.

3 ♦ Select the index you want to deactivate.

4 ♦ Choose the MakeInactive option from the menu at the bottom of the window. Progress prompts you to verify that you want to make the selected index inactive.

5 ♦ Choose Yes to deactivate the index.

6 ♦ Choose Exit to apply the change. Progress displays an alert box asking if you want to apply or undo the changes to the database, or if you want to make other changes.

7 ♦ Choose Apply Changes. Progress returns you to the Data Dictionary main window.

NOTE: For more information on indexes, see the chapter on Progress performance in the *Progress Database Administration Guide and Reference*.

4.6 Deleting an Index

Follow these steps to delete an index:

NOTE: You cannot delete the primary index. To delete the index that is currently the primary index, you must first make another index the primary index.

1 ♦ Choose Schema→ Index Editor. Progress alphabetically lists all the tables defined for your database.

2 ♦ Select a table. Progress lists the indexes defined for that table.

3 ♦ Select the index you want to delete. You cannot delete a primary index.

4 ♦ Choose the Delete option from the menu at the bottom of the window. Progress prompts you to verify that you want to remove this index.

5 ♦ Choose Yes to delete the index. Progress returns you to the Index Editor.

6 ♦ Choose Exit to apply the change. Progress displays an alert box asking if you want to apply or undo the changes to the database, or if you want to make other changes.

7 ♦ Choose Apply Changes. Progress returns you to the Data Dictionary main window.

4.7 Generating a Quick Index Report

Follow these steps to generate a quick index report:

1 ♦ Choose Database→ Reports→ Quick Index. Progress alphabetically lists all the tables defined for your database.

2 ♦ Select a table or type **ALL**, then press **RETURN**. Progress generates the report and displays the Quick Index Report dialog box.

3 ♦ Choose one of the following options:

a) Choose Switch Tables to generate a quick index report for another table. Progress alphabetically lists all the tables.

b) Choose Print to print the report or write it to a file. The Print Options dialog box appears:

- Specify the report destination. If you want to save the report to a file, specify the filename and whether to append the report to an existing file.
- Specify the page length, then choose OK. Progress sends the report to the specified output destination and returns you to the Quick Index Report dialog box.

4 ♦ Choose OK to return to the Data Dictionary window.

5

Sequence Tasks

This chapter describes the following tasks involved in creating and maintaining sequences:

- Creating a sequence
- Modifying a sequence
- Deleting a sequence
- Dumping sequence definitions
- Dumping sequence values
- Loading sequence values
- Generating a sequence report

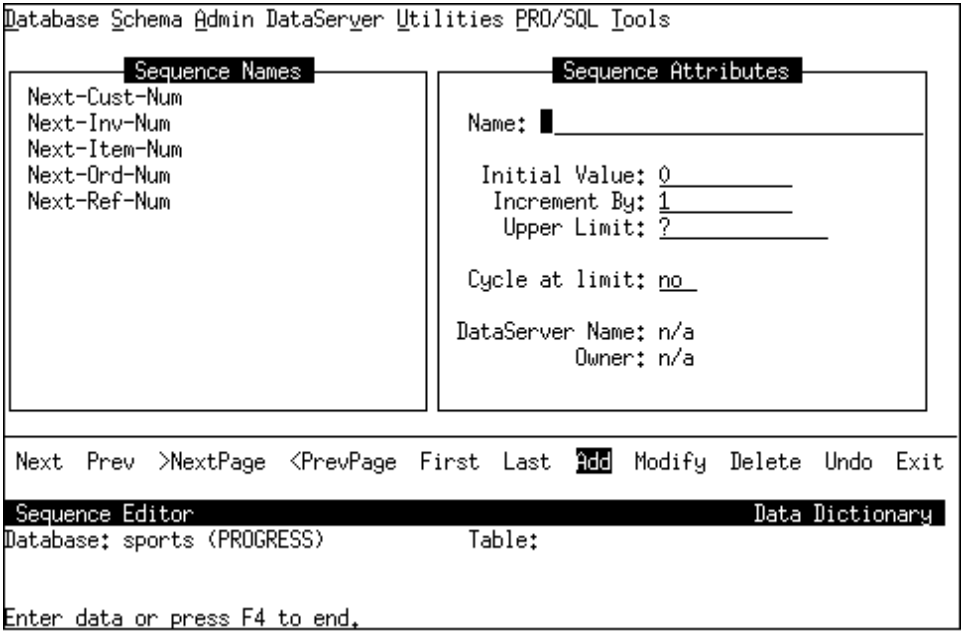
See the *Progress Database Administration Guide and Reference* for more information about dumping and loading sequences.

5.1 Creating a Sequence

Follow these steps to create a sequence:

1 ♦ Choose Schema→ Sequence Editor. Progress displays a list of the sequences defined for the working database and the attributes for the sequence that is selected.

2 ♦ Choose the Add option from the menu at the bottom of the window. Progress displays the following window:



3 ♦ Define the sequence attributes, then press **GO**. Progress then returns you to the Sequence Editor.

4 ♦ Choose Exit to apply the change. Progress displays an alert box asking if you want to apply or undo the changes to the database, or if you want to make other changes.

5 ♦ Choose Apply Changes. Progress returns you to the Data Dictionary main window.

5.2 Modifying a Sequence

Follow these steps to modify a sequence:

1 ♦ Make sure that the database containing the sequences you want to dump or load is the current working database.

2 ♦ Choose Schema→ Sequence Editor. Progress lists all the sequences defined for the working database and the attributes for the sequence that is selected.

3 ♦ Select the sequence you want to modify.

4 ♦ Choose the Modify option from the menu at the bottom of the window.

5 ♦ Edit the sequence definition, then press **GO**.

6 ♦ Choose Exit to apply the change. Progress returns you to the Data Dictionary main window.

5.3 Deleting a Sequence

Follow these steps to delete a sequence:

1 ♦ Make sure that the database containing the sequences you want to delete is the current working database.

2 ♦ Choose Schema→ Sequence Editor. Progress lists all the sequences defined for the working database and the attributes for the sequence that is selected.

3 ♦ Select the sequence you want to delete.

4 ♦ Choose the Delete option from the menu at the bottom of the window. Progress prompts you to verify that you want to remove the sequence.

5 ♦ Choose Yes to delete the sequence. Progress returns you to the Sequence Editor.

6 ♦ Choose Exit to apply the change. Progress returns you to the Data Dictionary main window.

5.4 Dumping Sequence Definitions

Follow these steps to dump sequence values:

1 ♦ Make sure that the database containing the sequence definitions you want to dump is the current working database.

2 ♦ Choose Admin→ Dump Data and Definitions→ Sequence Definitions. Progress displays the Dump Sequence Definitions dialog box.

3 ♦ Specify the filename or use the default value. The default filename is `_seqdefs.d`.

4 ♦ Specify the code page or use the default value, then choose OK.

After dumping the sequence values, Progress displays a status message and prompts you to continue.

5 ♦ Choose OK to return to the Data Dictionary main window.

5.5 Dumping Sequence Current Values

Follow these steps to dump sequence values:

1 ♦ Make sure that the database containing the sequence values you want to dump is the current working database.

2 ♦ Choose Admin→ Dump Data and Definitions→ Sequence Current Values. Progress displays the Dump Sequence Current Values dialog box.

3 ♦ Specify the filename or use the default value. The default filename is `_seqvals.d`.

4 ♦ Specify the code page or use the default value, then choose OK.

After dumping the sequence values, Progress displays a status message and prompts you to continue.

5 ♦ Choose OK to return to the Data Dictionary main window.

5.6 Loading Sequence Current Values

Progress uses a contents file that you specify to load sequence values into your database. Usually, this file is called `_seqvals.d` file.

To load sequence values into a database, follow these steps:

1 ♦ Make sure that the working database is the target database where you want to load the sequence values.

2 ♦ Choose Admin→ Load Data and Definitions→ Sequence Current Values. Progress prompts you for the filename from which you want to write the sequence values. The default filename is `_seqvals.d`.

3 ♦ Specify the filename or use the default value and choose OK. After the sequence values are loaded, Progress displays a status message and prompts you to continue.

4 ♦ Choose OK to return to the Data Dictionary main window.

5.7 Generating a Sequence Report

Follow these steps to generate a sequence report:

1 ♦ Make sure that the working database is the target database for which you want to generate the report.

2 ♦ Choose Database→ Reports→ Sequence. The Sequence Report dialog box appears and displays information about all sequences defined for the working database.

3 ♦ Choose the Print button if you want to print the report. The Print Options dialog box appears:

   a) Specify the report destination. If you want to save the report to a file, specify the filename and whether to append the report to an existing file.

   b) Specify the page length, then choose OK. Progress sends the report to the specified output destination and returns you to the Sequence Report dialog box.

4 ♦ Choose OK to return to the Data Dictionary window.

6

Trigger Tasks

There are two types of triggers: session and schema. You create session triggers as part of a Progress procedure and create schema triggers using the Data Dictionary. See the *Progress Programming Handbook* for information about creating session triggers.

There are two types of schema triggers:

- Table
- Field

The following chapter describes the following tasks involved in creating and maintaining schema triggers:

- Creating a schema trigger
- Modifying a schema trigger
- Deleting a schema trigger
- Generating a schema trigger report

NOTE: To generate r-code for schema trigger definitions, you must define the trigger procedure in the Data Dictionary tool, then compile and save the r-code with the Application Compiler.

6.1 Creating a Trigger

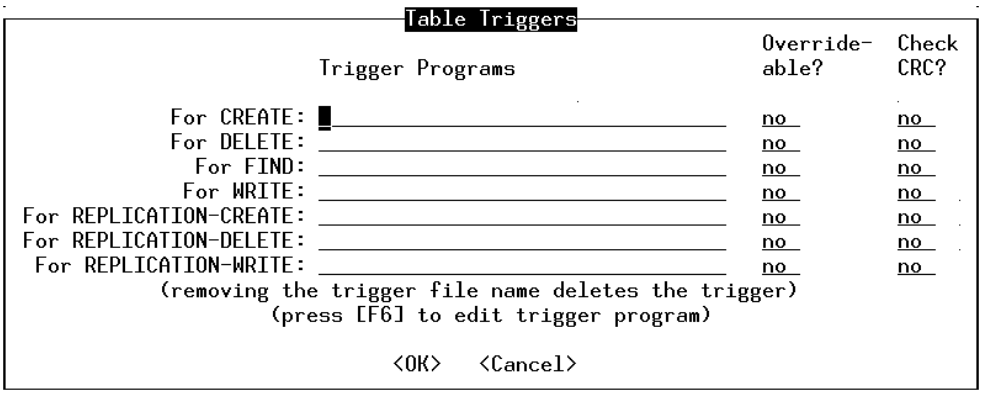
The following sections describe how to create table and field triggers.

6.1.1 Creating Table Triggers

Follow these steps to create a table trigger:

1 ♦ Choose Schema→ Add New Table or choose Schema→ Modify Table. If you are adding a new table, specify a table name. If you are modifying a table, Progress alphabetically lists all the tables defined for your database. Select the table for which you want to create the trigger.

2 ♦ Choose the Triggers option. Progress displays the Table Triggers dialog box:



3 ♦ Define the trigger attributes. For more information on the Table Triggers dialog box and the trigger attributes, see Chapter 9, "Schema Menu Reference."

4 ♦ Specify whether a session trigger can override this schema trigger. For more information about session and schema triggers, see the chapter on database triggers in the *Progress Programming Handbook*.

5 ♦ Specify whether Progress checks the cyclical redundancy check (CRC) code for the trigger.

6 ♦ Press **PUT** to enter the trigger code. Progress displays the Trigger Code dialog box.

7 ♦ Enter the trigger code.

8 ♦ Choose OK to return to the Table Triggers dialog box.

9 ♦ Choose OK to return to the Add Table or Modify Table window.

6.1.2 Creating a Field Trigger

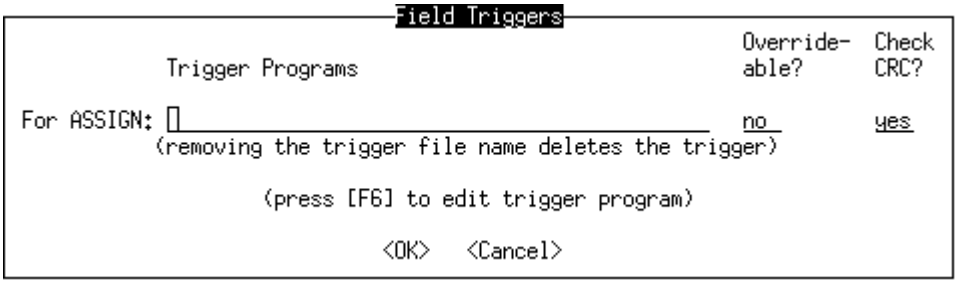
Follow these steps to create a field trigger for an existing field:

1 ♦ Choose Schema→ Field Editor. Progress alphabetically lists all the tables defined for your database.

2 ♦ Select the table that contains the field for which you want to create the trigger. The Field Editor window appears.

3 ♦ Choose the Triggers option at the bottom of the window.

4 ♦ Select the field for which you want to define the trigger. The Field Triggers dialog box appears:



5 ♦ Specify the name of the trigger procedure, then press **PUT** to enter the trigger code. The Trigger Editor appears.

6 ♦ Specify whether a session trigger can override the schema trigger. For more information about session and schema triggers, see the chapter on database triggers in the *Progress Programming Handbook*.

7 ♦ Specify whether Progress verifies the cyclical redundancy check (CRC) code.

8 ♦ Choose OK to create the trigger and return to the Field Editor.

6.2 Modifying a Trigger

The following sections describe how to modify table and field triggers.

6.2.1 Modifying a Table Trigger

Follow these steps to modify a table trigger:

1 ♦ Choose Schema→ Modify Table. Progress alphabetically lists all the tables defined for your database.

2 ♦ Select the table for which you want to modify the trigger. The Modify Table window appears.

3 ♦ Choose the Triggers button. The Table Triggers dialog box appears.

4 ♦ Select the trigger you want to modify.

5 ♦ Press **PUT** to enter the trigger code. Progress displays the Trigger Code dialog box.

6 ♦ Edit the trigger code.

7 ♦ Choose OK to return to the Table Triggers dialog box.

8 ♦ Choose OK in the Table Triggers dialog box to return to the Modify Table window.

6.2.2 Modifying a Field Trigger

Follow these steps to modify a field trigger:

1 ♦ Choose Schema→ Field Editor. Progress alphabetically lists all the tables defined for your database.

2 ♦ Select the table for which you want to modify the trigger. The Field Editor appears.

3 ♦ Choose the Triggers option at the bottom of the window.

4 ♦ Select the field for which you want to modify the trigger. The Field Triggers dialog box appears.

5 ♦ Specify the name of the trigger procedure, then press **PUT** to enter the trigger code. The Trigger Code window appears.

6 ♦ Edit the trigger code.

7 ♦ Choose OK to return to the Field Triggers dialog box.

8 ♦ Choose OK in the Field Triggers dialog box to return to the Field Editor.

6.3 Deleting a Trigger

The following sections describe how to delete table and field triggers.

NOTE: When you delete a trigger from the Data Dictionary, you do not delete the trigger procedure itself.

6.3.1 Deleting a Table Trigger

Follow these steps to delete a table trigger:

1 ♦ Choose Schema→ Modify Table. Progress alphabetically lists all the tables defined for your database.

2 ♦ Select the table for which you want to delete the trigger. The Modify Table window appears.

3 ♦ Choose the Triggers button. The Table Triggers dialog box appears.

4 ♦ Delete the trigger filename for the trigger you want to remove.

5 ♦ Choose OK to return to the Modify Table window.

6.3.2 Deleting a Field Trigger

Follow these steps to delete a field trigger:

1 ♦ Choose Schema→ Field Editor. Progress alphabetically lists all the tables defined for your database.

2 ♦ Select the table for which you want to delete the trigger. The Field Editor appears.

3 ♦ Choose the Triggers option at the bottom of the window.

4 ♦ Choose the field for which you want to delete the trigger. The Field Triggers dialog box appears.

5 ♦ Delete the trigger filename for the trigger you want to remove.

6 ♦ Choose OK in the Field Triggers dialog box to return to the Field Editor.

6.4 Generating a Trigger Report

Follow these steps to generate a trigger report:

1 ♦ Choose Database→ Reports→ Trigger. The Trigger Report dialog box appears.

2 ♦ Choose the Print button if you want to print the report. The Print Options dialog box appears.

3 ♦ Specify the report destination. If you want to save the report to a file, specify the filename and whether to append the report to an existing file.

4 ♦ Specify the page length, then choose OK. Progress sends the report to the specified output destination and returns you to the Trigger Report dialog box.

5 ♦ Choose OK to return to the Data Dictionary main window.

7

SQL-89 Tasks

You perform most of the SQL-89 tasks with the PRO/SQL-89 menu. This chapter describes the following SQL-89 tasks:

- Generating an SQL-89 program
- Dumping the definitions for SQL-89 views
- Loading the definitions for SQL-89 views
- Generating an SQL-89 view report

7.1 Generating an SQL-89 Program

You can generate SQL-89 DDL programs that contain CREATE VIEW and CREATE TABLE statements equivalent to those originally used to define the selected view or table.

7.1.1 Generating CREATE VIEW Statements

Follow these steps to generate a CREATE VIEW statement for a view:

1 ♦ Choose PRO/SQL→ Dump as CREATE VIEW Statement. Progress alphabetically lists all the views defined for your database.

2 ♦ Select a view or type **ALL**.

3 ♦ Specify the name of the output file. The default is the name of the view, with a .p extension.

4 ♦ Specify whether Progress should use a period (.) or a semi-colon (;) as the symbol for the termination of the CREATE VIEW statement, then choose OK.

After generating the file, Progress prompts you to continue.

5 ♦ Choose OK to return to the Data Dictionary main window.

7.1.2 Generating CREATE TABLE Statements

Follow these steps to generate a CREATE TABLE statement for a table:

1 ♦ Choose PRO/SQL→ Dump as CREATE TABLE Statement. Progress alphabetically lists all the tables defined for your database.

2 ♦ Select a table or type **ALL**.

3 ♦ Specify the output filenames for the CREATE TABLE and CREATE INDEX statements. If you do not specify a filename for the CREATE INDEX statement, Progress writes the CREATE INDEX statement to the CREATE TABLE statement output file.

4 ♦ Select the type of SQL-89 you want to use, then choose OK.

After generating the file, Progress prompts you to continue.

5 ♦ Choose OK to return to the Data Dictionary main window.

7.2 Dumping the Definitions for SQL-89 Views

Follow these steps to dump the definitions for the SQL-89 views:

1 ♦ Choose Admin→ Dump Data and Definitions→ SQL Views. Progress prompts you for the file to which you want to write the definitions for the views.

2 ♦ Specify the filename or accept the default. The default filename is `_view.d`.

3 ♦ Specify the code page, then choose OK.

After dumping the view definitions to the specified file, Progress displays a status message and prompts you to continue.

4 ♦ Choose OK to return to the Data Dictionary window.

7.3 Loading the Definitions for SQL-89 Views

Follow these steps to load the definitions for SQL-89 views:

1 ♦ Choose Admin→ Load Data and Definitions→ SQL Views. Progress prompts you for the file from which you want to write the user file contents.

2 ♦ Specify the filename or accept the default. The default filename is `_view.d`. After loading the view definitions from the specified file, Progress displays a status message and prompts you to continue.

3 ♦ Choose OK to return to the Data Dictionary main window.

7.4 Generating an SQL-89 View Report

This option displays information about all PRO/SQL views defined for the working database. Views of non-Progress databases are treated as tables. To display information for these views, use the table reports option. See Chapter 2, "Table Tasks," for more information on generating a table report.

Follow these steps to generate a PRO/SQL view report:

1 ♦ Choose PRO/SQL→ PRO/SQL View Report or choose Database→ Reports→ PRO/SQL View Report. Progress alphabetically lists all the views defined for your database.

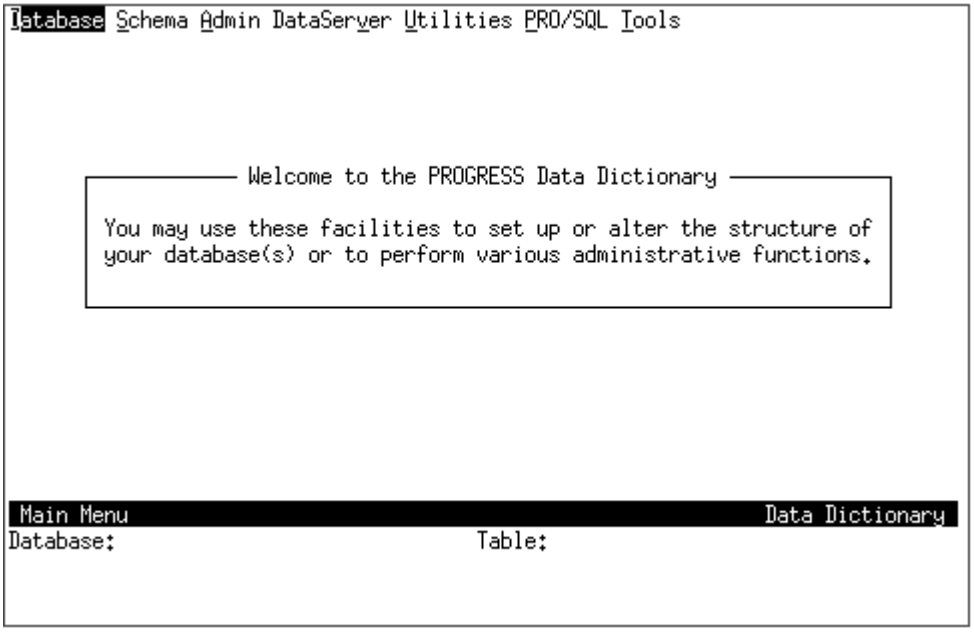
2 ♦ Select a view or type **ALL**. Progress generates the report.

3 ♦ Choose the Print button if you want to print the report or write it to a file. The Print Options dialog box appears:

a) Specify the report destination. If you want to save the report to a file, specify the filename and whether to append the report to an existing file.

b) Specify the page length, then choose OK. Progress sends the report to the specified output destination and returns you to the SQL View Report dialog box.

4 ♦ Choose OK to return to the Data Dictionary main window:



8

Database Menu Reference

This chapter describes the menu options of the Progress Data Dictionary. The Data Dictionary lets you create and modify database schema, including tables, fields, sequences, indexes, and triggers. You can also generate schema reports and perform database administration tasks.

For information on how to use the Data Dictionary to perform tasks, see the previous chapters of this book and the *Progress Database Administration Guide and Reference*.

8.1 Data Dictionary Menu Bar

The following sections describe the pull-down menus and options available from the menu bar in the Data Dictionary. Figure 8–1 shows the Data Dictionary menu bar.



Figure 8–1: Data Dictionary Menu Bar

Table 8–1 briefly describes the Data Dictionary menu bar options.

Table 8–1: Data Dictionary Menu

Database	
Database	Manages database creation, connection, and disconnection, and generates schema reports. Also exits the Data Dictionary.
Schema	Modifies database structure, including defining tables, fields, indexes, sequences, and triggers.
Admin	Dumps and loads data and definitions and defines security for database access.
DataServer	Accesses utilities to view and maintain the schema images for non-Progress databases.
Utilities	Creates parameter files, freezes/unfreezes tables, prepares data for dumping and loading, and generates include files.
PRO/SQL	Performs SQL-related tasks.
Tools	Accesses other Progress tools.

The menu bar provides access to all the available options. However, if you do not have a database connected, you can only access the options listed in Table 8–2.

Table 8–2: Options Available Without a Connected Database

Menu Option	Pull-Down Menu Option	Task
Database	Create...	Creates a new database.
	Connect...	Connects to an existing database.
	Exit	Exits the Data Dictionary.
Utilities	Editor for Parameter Files...	Creates or edits a parameter file.
	Quoter Functions	Formats data files so that Progress can read the data into a Progress database.
	Information ...	Displays session information.
Tools	All options	Accesses other Progress tools.

You must connect to a database to access the other menu options.

8.2 Database Menu

The Database Menu lets you:

- Switch to a different working database
- Create new Progress databases
- Connect or disconnect databases
- Generate schema reports
- Exit from the Data Dictionary

How to connect and disconnect databases is important information to know if you are working with multiple databases. For more information, see the *Progress Programming Handbook*. Table 8–3 describes the menu that appears when you choose this option. More detailed descriptions of the options follow the table.

Table 8–3: Database Menu

Select Working Database...	Specifies the working database.
Create...	Creates a new Progress database.
Connect...	Connects to an existing database.
Disconnect...	Disconnects a connected database.
Reports→	Generates reports about various sets of schema information.
Exit	Exits from the Data Dictionary.

8.2.1 Database→ Select Working Database

Choose Database→ Select Working Database to specify the working database if you have more than one database connected. When you choose this option, the Select Working Database dialog box shown in Figure 8–2 appears.

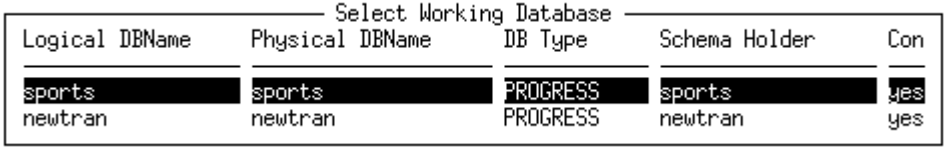


Figure 8–2: Select Working Database Dialog Box

Progress lists the available databases and basic database information. If you name a database when you start Progress with the PRO utility, that database is connected and is the working database.

8.2.2 Database→ Create

Choose Database→ Create to create a Progress database. When you choose this option, the Create Database dialog box appears as shown in Figure 8–3.

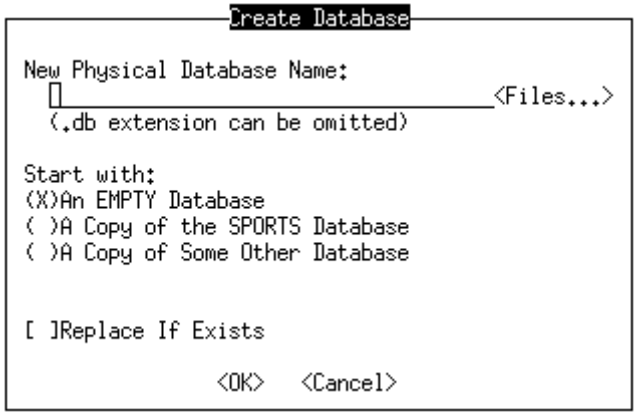


Figure 8–3: Create Database Dialog Box

This dialog box contains the following user-interface elements:

New Physical Database Name

Specifies the actual name for the database on a disk. This name can contain an explicit path or it can be relative to your propath.

Files

Displays a list of available options for its corresponding field. For example, when you choose the Files button next to the New Physical Database Name field, the Find Database File dialog box shown in Figure 8–4 appears.

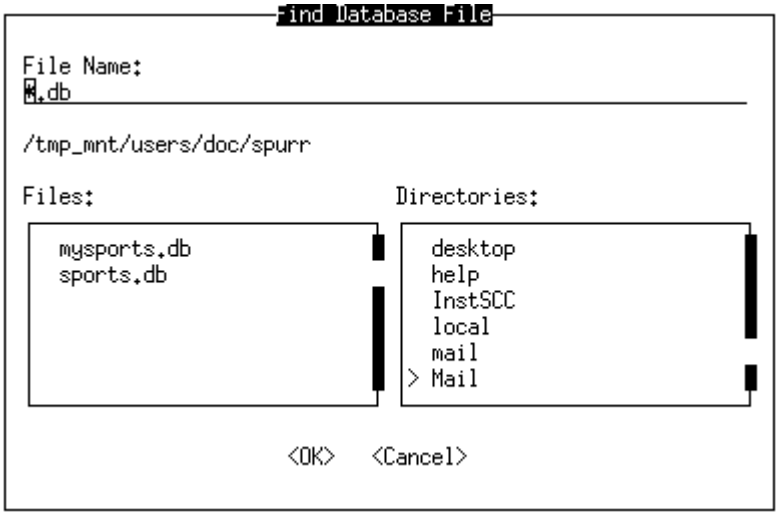


Figure 8–4: Find Database File Dialog Box

Start with

Specifies the type of database you want to create:

- **An EMPTY Database** — Specifies that you want to create an empty database.
- **A Copy of the SPORTS Database** — Specifies that you want to create a copy of the Progress sports database.
- **A Copy of Some Other Database** — Specifies that you want to create a copy of an existing database. If you choose this button, you must specify the name of the database you want to copy in the field beneath this option.

Replace If Exists

Indicates that if you specify the name of an existing database, you want to replace the existing database with the one you are creating.

NOTE: When you create a database with the PROSTRCT command, you can specify a database block size that is a multiple of 1024K, from 1024K to 8192K. Specifying a block size that matches the operating system block size might improve performance.

Raw Partition Support

You can use the PROSTRCT CREATE command to create a database with raw partitions. Storing portions of a database on raw partitions can help you avoid problems associated with the UNIX file system, such as asynchronous I/O, large file sizes, and security. Creating a database with raw partitions also enables you to avoid using the `sync ( )` system call.

8.2.3 Database→ Connect

If Progress creates the database successfully, the Database Connect dialog box shown in Figure 8–5 appears.

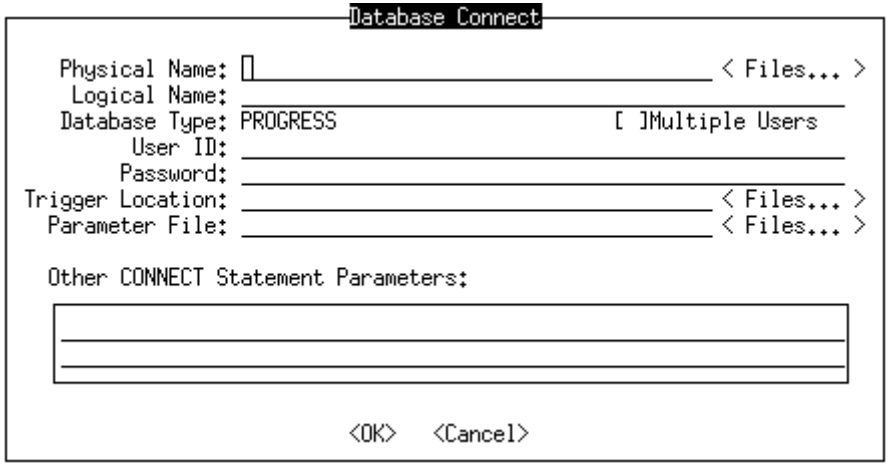


Figure 8–5: Database Connect Dialog Box

This dialog box contains the following user-interface elements:

Physical Name

Specifies the actual name of the database on a disk. This name can contain an explicit path or it can be relative to your propath.

Logical Name

Specifies the database name that references a connected physical database. For example, if the physical database name is `mydb1.db`, the default logical database name is `mydb1`. For more information on logical database names, see the *Progress Programming Handbook*.

Database Type

Specifies the database type. By default, the database type is Progress. The only possible value for this release is Progress.

Multiple Users

Specifies if you want multiple users to be able to access this database at the same time. The default is single-user. An x indicates multiple users can access the database. Before you can connect to a multi-user database, you must start a server for it using the PROSERVE utility.

NOTE: You **must** select this option if you are running a multi-user server and you want to connect to a self-service client.

Userid

Identifies your user ID. Whether you enter a user ID and password depends on how you initially created the database. If you created a new database from the empty or sports database, then you don't have to enter any values for user ID or password because, by default, the security for these databases is not enabled. However, if you created a database, for example, mydb, from a copy of another database that has a list of user IDs and passwords in the user list, then you must specify a valid user ID and password to connect to mydb.

Progress checks the entries, and if it finds a match for them in the user list, it assigns the user ID to the Progress session. Otherwise, Progress does not allow you to connect to mydb. For more information on user ID, and passwords, see the *Progress Database Administration Guide and Reference*.

Password

Identifies your password.

Trigger Location

Specifies the directory or Progress r-code library containing the trigger code for the database.

Parameter File

Specifies the parameter filename that contains the startup parameters for the database. The parameter file can include any number of startup parameters to use when you run Progress with this database. For more information about parameter files, see the *Progress Database Administration Guide and Reference* and the *Progress Client Deployment Guide*.

Other CONNECT Statement Parameters

Specifies any other startup parameters for the database that are not included in the parameter file. For example, enter any parameters you want to use that are not defined by this dialog box (or by the parameter file, if you included one in the Parameter File area).

Internally, the Data Dictionary constructs and executes a CONNECT statement using the information you supply. Therefore, any rules that apply to the CONNECT statement also apply to database connections using the Progress Data Dictionary. See the *Progress Programming Handbook* for more information about the CONNECT statement. For more information about connection parameters, see the *Progress Startup Command and Parameter Reference*.

8.2.4 Database→ Disconnect

Choose this option to disconnect a database. When you have more than one database connected, Progress displays the Select Database to Disconnect dialog box. Select the database you want to disconnect. Progress prompts you to verify that you want to disconnect the database. If you have two databases connected and you disconnect the working database, Progress automatically switches to the other database. If you have more than two databases connected and you disconnect the working database, Progress displays the Select Working Database dialog box and lets you select the new working database.

8.2.5 Database→ Reports

Choose Database→ Reports to generate reports about the schema in the database. Table 8–4 describes the submenu that appears when you choose this option.

Table 8–4: Reports Menu *(1 of 2)*

Report	
Detailed Table	Displays details about one or all tables in this database, including field, sequence, and index information.
Quick Table	Displays a summary of all the tables in this database.
Quick Field	Displays field information for one or all tables defined in the working database.
Quick Index	Displays index information for one or all tables defined in the working database.

Table 8–4: Reports Menu *(2 of 2)*

Report	
PRO/SQL View	Displays PRO/SQL view information for all the views defined in the working database.
Sequence	Displays sequences information for all the sequences defined in the working database.
Trigger	Displays trigger information for all the triggers defined in the working database.
User	Displays the current database users, user IDs, and whether each user has a password. You specify this information with the Data Dictionary.
Table Relations	Displays the relations for one or all tables defined in the working database.
Storage Areas	Displays the storage area information for all storage areas defined in the working database.

Reports→ Detailed Table

Choose Reports→ Detailed Table to generate information about the structure of the selected table or of all the tables in the working database. This option creates a report that displays all information about the tables, including information about fields and indexes defined for the tables.

When you choose this option and you select a table, Progress displays the Report Options dialog box as shown in Figure 8–6.

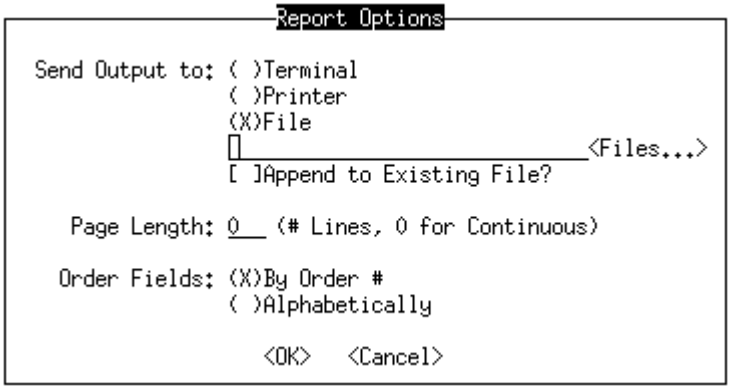


Figure 8–6: Report Options Dialog Box

This dialog box has the following user-interface elements:

Send Output to

Lets you send output to the screen, the default printer for your computer system, or a file.

If you choose to send output to a file, specify the filename and whether to append the report information to an existing file. Choose the Files button to display a list of existing files.

Page Length

Specifies the number of lines per page. The default for the printer is 60 lines; for the terminal and a file it is 0 (continuous).

Order Fields

Specifies whether to order the fields for the report by order number or alphabetically.

Figure 8–7 shows a portion of a detailed table report for all the tables in the sports database.

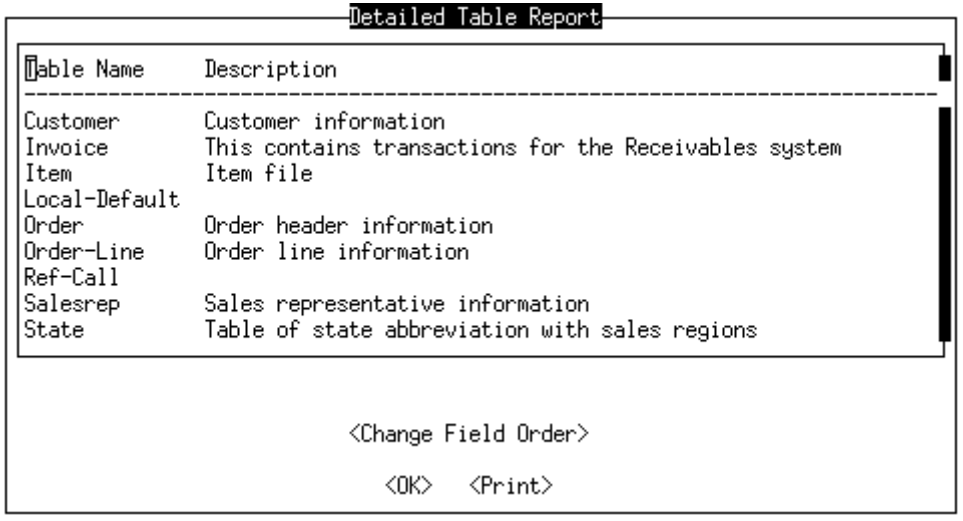


Figure 8–7: Detailed Table Report Dialog Box

Reports→ Quick Table

Choose Reports→ Quick Table to display summary information about all the tables in the working database. For each table Progress lists the table name, storage area number, dump name, table flags, field count, index count, and table label. Figure 8–8 shows an example of a quick table report for the sports database.

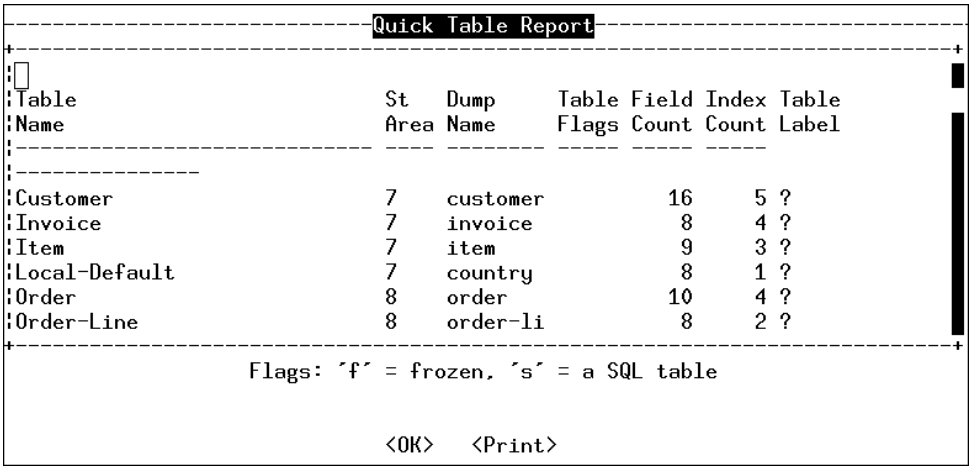


Figure 8–8: Quick Table Report Dialog Box

Reports→ Quick Field

Choose Reports→ Quick Field to display information for the structure of fields in one or all tables in the working database. For each field, Progress lists the order number, name, data type, flags, and format.

The Data Type column lists the first four characters of the field's data type. If the type is decimal, the report includes the number of stored decimals digits (for example, deci–2). If the field is an array, the report includes the extent (for example, char[10]). The report shows a decimal array field as deci–2[5]. The Flags column indicates whether the field is case sensitive, mandatory, an index component, or a view component. Figure 8–9 shows an example of a quick field report for the customer table in the sports database.

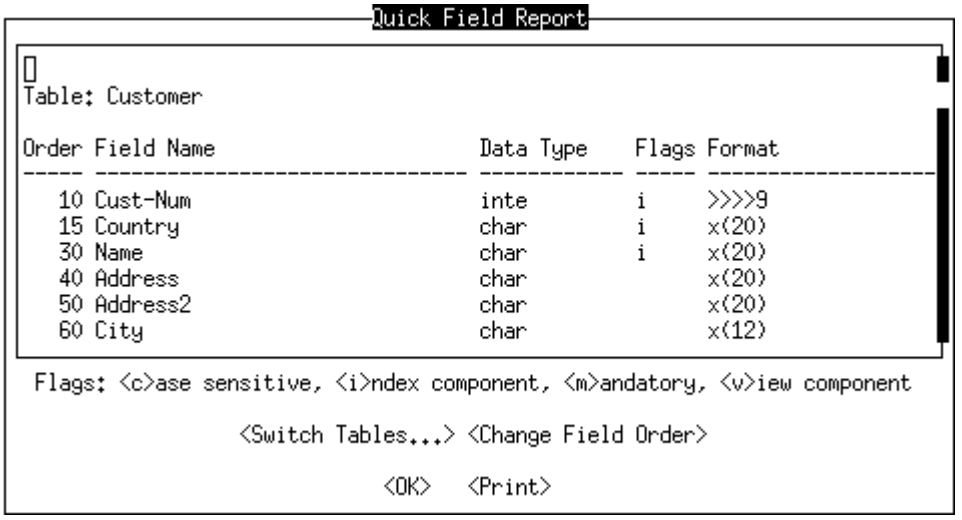


Figure 8–9: Quick Field Report Dialog Box

The Quick Field Report window has the following user-interface elements:

Switch Tables

Returns you to the list of tables for the working database.

Change Field Order

Changes the sort order of the fields to either alphabetical by name or numeric by order.

Print

Lets you print the report or write it to a file.

Reports→ Quick Index

Choose Reports→ Quick Index to display information about the indexes defined in one or all tables in the working database. For each table, the report lists the index flags, name of the index, storage area number, and fields that define the index. The Flags column indicates whether the indexes are primary, unique, word, abbreviated, or inactive. A plus (+) or minus (-) sign next to the field name indicates if the index is ascending or descending. Figure 8–10 shows an example of a quick index report for the customer table of the sports database.

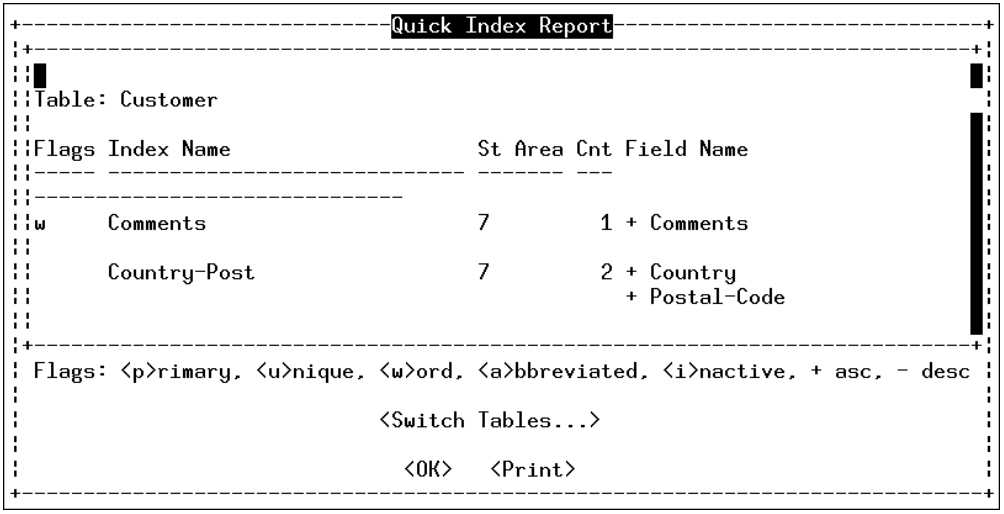


Figure 8–10: Quick Index Report Dialog Box

The Quick Index Report window has the following user-interface elements:

Switch Tables

Returns you to the list of tables for the working database.

Print

Lets you print the report or write it to a file.

Reports→ PRO/SQL View

Choose Reports→ PRO/SQL View to display information about all SQL views defined for the working database. For each view, it shows the name of the view, whether each view can be updated, and if a GROUP BY phrase or a check option was used. It also shows the base table name, WHERE clause criteria, and view definition. Figure 8–11 shows an example of an SQL View Report for the customer table of the sports database.

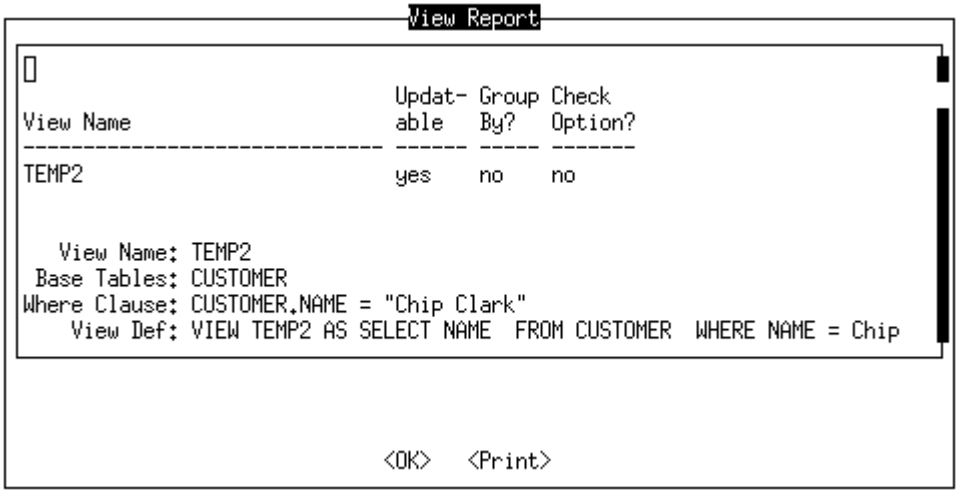


Figure 8–11: PRO/SQL View Report Dialog Box

Choose the Print button to print the report or to write it to a file.

Reports→ Sequence

Choose Reports→ Sequence to display information about all sequences defined in the current database. For each sequence, it shows the name of the sequence, its initial value, increment value, minimum or maximum value, and whether the sequence cycles or stops at the maximum value. Figure 8–12 shows an example of a sequence report for the sports database.

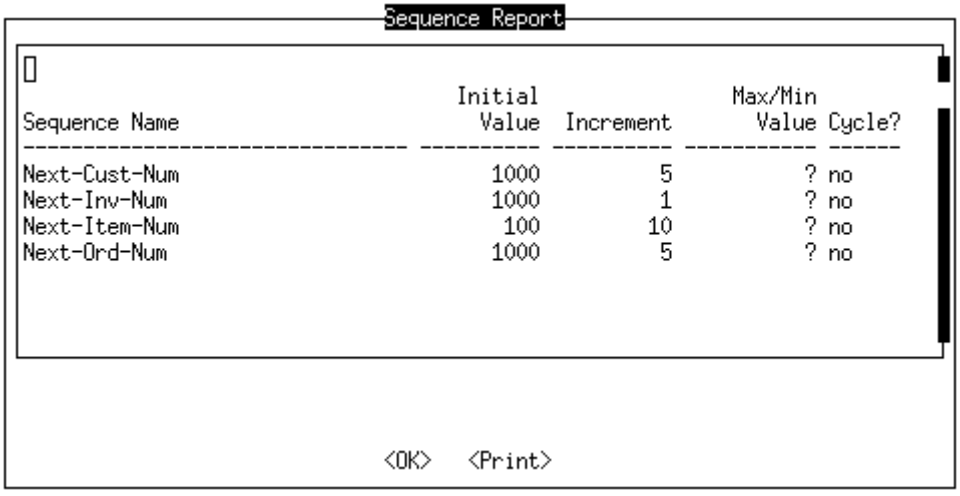


Figure 8–12: Sequence Report Dialog Box

Choose the Print button to print the report or to write it to a file.

Reports→ Trigger

Choose Reports→ Trigger to list all tables and fields in the report that have triggers defined for them. It lists the table or field name for which the trigger is defined, type of event that causes the trigger to fire, whether the Dictionary checks the cyclical redundancy check (CRC) code for the trigger procedure, trigger flags, and the name of the procedure. The Flags column indicates if the trigger is overridable, has a mismatched CRC code, or has no r-code available. Figure 8–13 shows an example of a trigger report for the sports database.

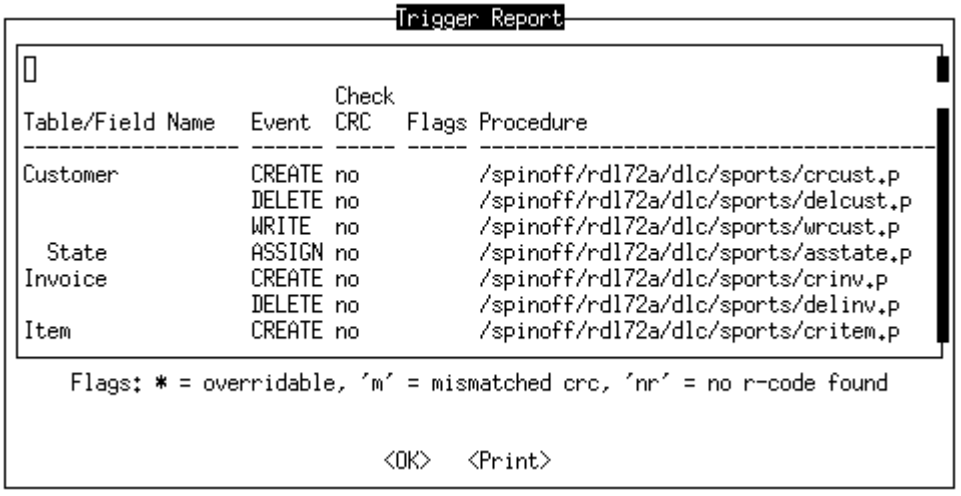


Figure 8–13: Trigger Report Dialog Box

Choose the Print button to print the report or to write it to a file.

Reports→ User

Choose Reports→ User to display the list of users who have permission to use the working database. This information is stored in the user list in the `_User` file. Figure 8–14 shows an example of a user report.

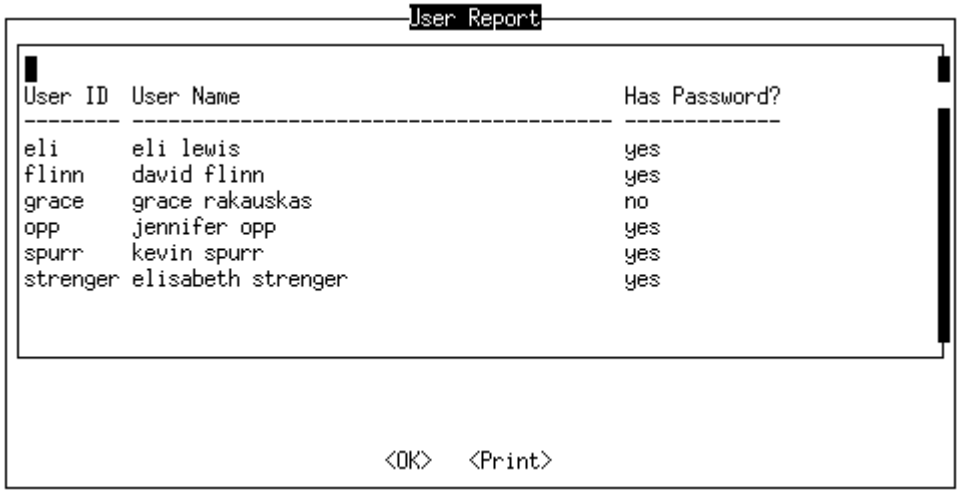


Figure 8–14: User Report Dialog Box

Choose the Print button to print the report or to write it to a file.

Reports→ Table Relations

Choose Reports→ Table Relations to list the implicit join relations for one or all tables in the working database. Figure 8–15 shows an example of a Table Relations Report for the customer table of the sports database.

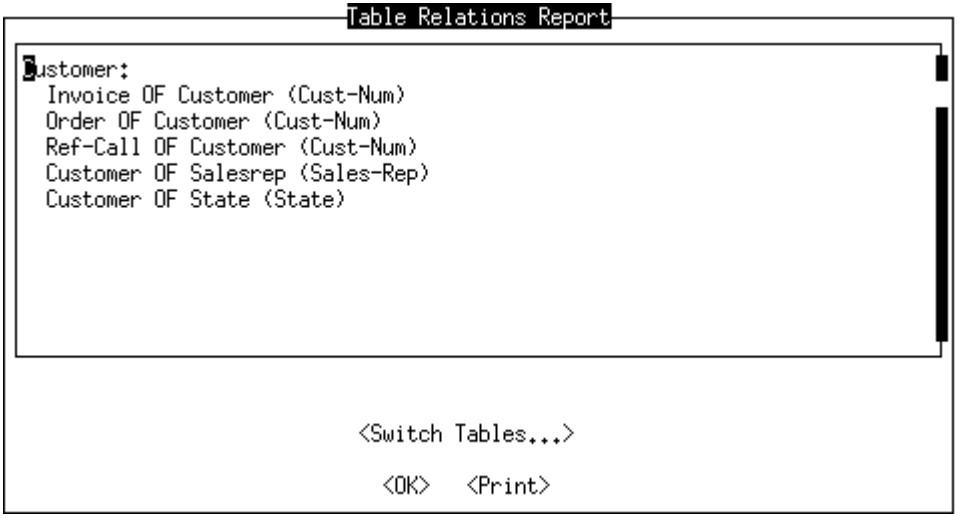


Figure 8–15: Table Relations Report Dialog Box

The Table Relations Report window has the following user-interface elements:

Switch Tables

Returns you to the list of tables for the working database.

Print

Lets you print the report or write it to a file.

Reports→ Storage Areas

Choose Reports→ Storage Areas to list the storage areas in the working database. The report lists storage area number, type, block size, total extents, and area name. Figure 8–16 shows an example of a Storage Areas Report for all tables of the sports database.

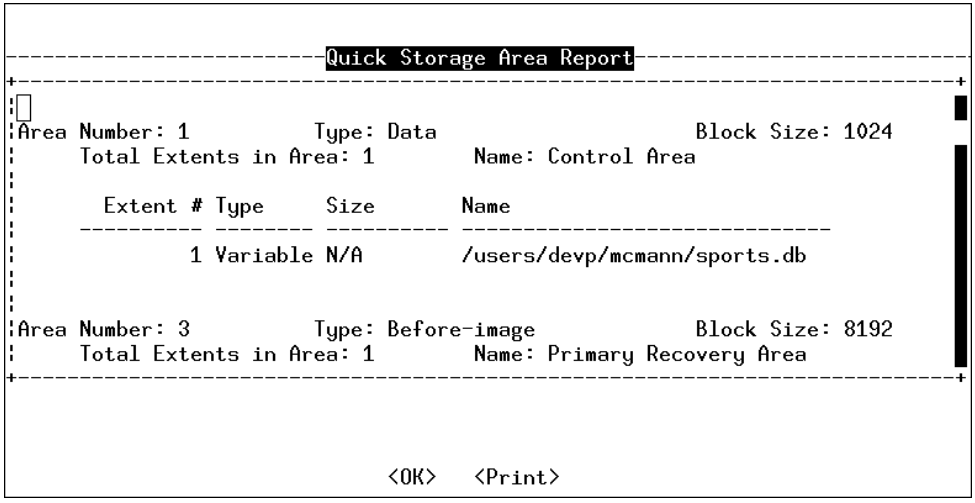


Figure 8–16: Quick Storage Areas Report Dialog Box

The Quick Storage Area Report window has the following user-interface element:

Print

Lets you print the report or write it to a file.

8.2.6 Database→ Exit

Choose this option to close the Data Dictionary. You return to where you started the Dictionary. For example, if you started the Dictionary from the Procedure Editor, you return there.

When you have more than one database connected and you disconnect the working database, Progress automatically switches to the next connected database on the list.

9

Schema Menu Reference

This chapter describes the Schema menu and dialog boxes. Choose schema menu commands to modify the structure or schema of a database.

9.1 Schema Menu

The *schema* of the database is the definition of its structure: the tables it contains, the fields within the tables, sequences, etc. In addition to database structure, Progress database schemas contain validation expressions and validation messages.

Choose the Schema options to add, update, or delete table, field, index, or sequence definitions. Table 9–1 lists the Schema menu options and gives a brief description of each. More detailed descriptions of the options follow the table.

Table 9–1: Schema Menu

Schema	
Modify Table...	Changes information defined for an existing table.
Add New Table...	Creates a new table definition.
Delete Table(s)...	Removes one or more table definitions.
Field Editor...	Adds, alters, or removes fields from a table definition.
Reorder Fields...	Resequences the fields into alphabetical order, or resets the order numbers of the fields.
Global Field Name Change...	Changes a field name throughout the database.
Index Editor...	Adds, alters, or removes indexes in a table.
Sequence Editor...	Adds, alters, or removes sequences in a table.

9.1.1 Schema→ Modify Table

Choose Schema→ Modify Table to change the structure of a table.

When you choose this option, Progress alphabetically lists all the tables defined for your database. After you select a table, a window similar to the one shown in Figure 9–1 appears.

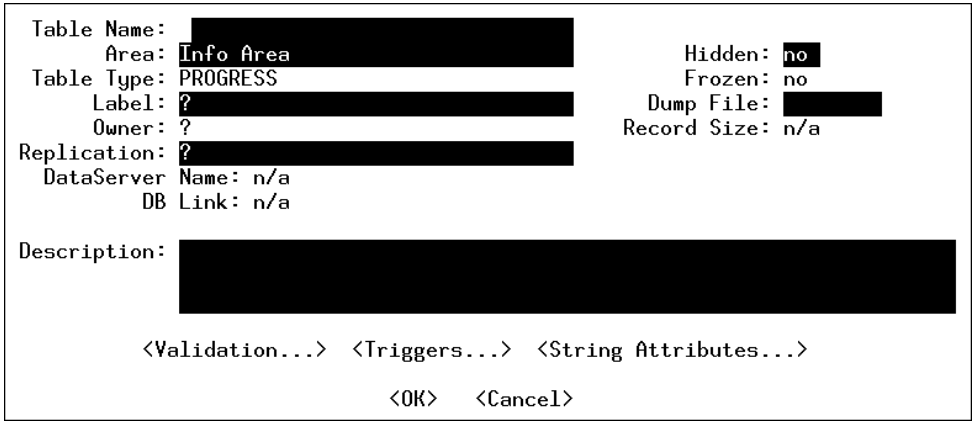


Figure 9–1: Modify Table Window

Choose the Field Editor button to edit fields for the selected table. See the "Schema→ Field Editor" section later in this chapter for more information.

This window contains the following user-interface elements:

Table Name

Specifies the name for the table. Table names can be up to 32 characters long and can consist of alphabetic characters, digits, and the characters $, &, #, %, -, and _. In addition, table names must begin with a letter (A–Z or a–z). You cannot use reserved Progress keywords as table names. Table names are not case sensitive.

Area

Displays the name of the storage area. The Area Name is assigned to a table when it is created. Because the Table Name cannot be updated, this field is disabled.

Table Type

Displays the table type. For this release, the only available table type is PROGRESS.

Label

Specifies the label used in error messages.

Owner

Displays the user ID of the table owner. This information is necessary for manipulating certain dataservers.

Replication

Specifies the name of the procedure that performs the replication for this table. For more information about replication, see the chapter on data replication in the *Progress Database Administration Guide and Reference*.

DataServer Name

Specifies the table name in the appropriate dataserver database if the table is not a Progress table.

DB Link

Specifies the link name for the distributed ORACLE database.

Description

Describes the table contents.

Hidden

Specifies if the table is a hidden table. The default value is no.

Frozen

Specifies if the table is frozen. You can freeze or unfreeze a table using the Freeze/Unfreeze option from the Utilities menu. When a table is frozen, you can only view its properties. The actual record data is not frozen, only the data definitions.

Dump File

Specifies the file to which you want to dump the table contents. The name must be unique among the other tables in the database and can be up to eight characters long. The default filename is the table name. If the table name is longer than eight characters, Progress truncates it to the first eight characters.

Record Size

Specifies the size of the record used for dataservers with fixed-length records.

The following sections describe the options that appear at the bottom of the window:

- Validation
- Triggers
- String Attributes

Table Validation Option

Choose the Validation option in the Modify Table dialog box to specify validation criteria and a message for record deletions.

NOTE: You should use table validation for backward compatibility for procedure-driven applications. For event-driven programs, Progress Software Corporation recommends that you use table delete triggers instead of table validation.

If you choose the validation option, a dialog box similar to the one shown in Figure 9–2 appears.

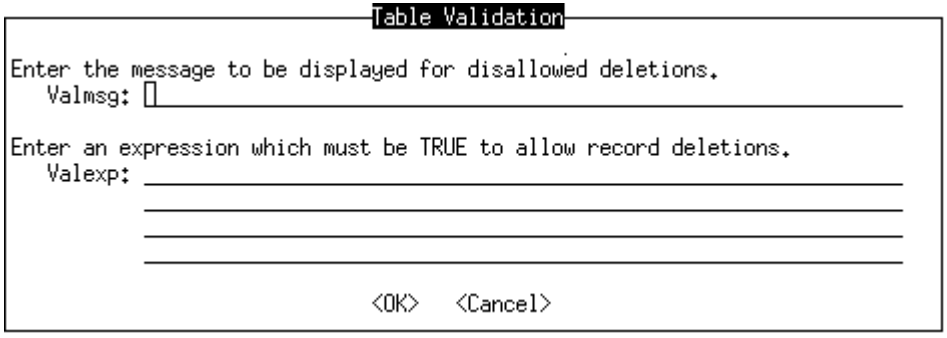


Figure 9–2: Table Validation Dialog Box

This window contains the following user-interface elements:

Valmsg

Specifies the message Progress displays when a user tries an invalid deletion. There is no limit to the number of characters you can enter, but a message up to 63 characters long can fit onto the two message lines without scrolling.

Valexp

Specifies the criteria for valid deletions. For example, in the customer table, to restrict deleting customers to those customers who have no outstanding orders, enter the following criteria:

```
NOT (CAN-FIND(FIRST order OF customer))
```

Table Triggers Option

Choose the Table Triggers option in the Modify Table dialog box to define schema triggers for a table. If you choose the Table Triggers option, a window similar to the one shown in Figure 9–3 appears.

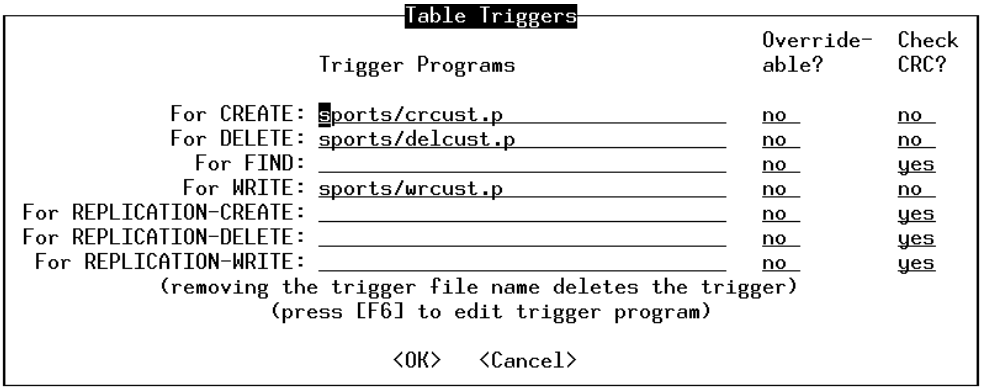


Figure 9–3: Table Triggers Dialog Box

The Table Triggers dialog box contains the following user-interface elements:

Overridable?

Specifies whether a session trigger can override the schema trigger. The default value is no. See the *Progress Programming Handbook* for more information about schema and session triggers.

Check CRC?

Specifies whether Progress verifies the cyclical redundancy check (CRC) code. The default value is no. When Progress checks the code, it compares the stored CRC value of the procedure to the CRC of the trigger when it fires.

Choose the database event for the trigger you want to create. Table 9–2 lists the available database events.

Table 9–2: Table Trigger Events

Event	Description
CREATE	Specifies that the trigger executes every time Progress executes a CREATE or INSERT statement for the table you are modifying.
DELETE	Specifies that the trigger executes every time Progress executes a DELETE statement for the table you are modifying.
FIND	Specifies that the trigger executes every time Progress reads a record in the table you are modifying using a FIND statement or a FOR EACH loop. Only records that satisfy the full search conditions (such as a WHERE clause) cause a FIND trigger to execute. The CAN-FIND statement does not cause a FIND trigger to execute.
WRITE	Specifies that the trigger executes every time Progress validates a record (if any change took place). Progress automatically validates a record when it releases it.
REPLICATION-CREATE	Specifies the trigger that executes immediately following the execution of the CREATE trigger, if specified. The purpose of the trigger is to replicate the creation of records to another database.
REPLICATION-DELETE	Specifies the trigger that executes immediately following the execution of the DELETE trigger, if specified. The purpose of the trigger is to replicate the deletion of records to another database.
REPLICATION-WRITE	Specifies the trigger that executes immediately following the execution of the WRITE trigger, if specified. The purpose of the trigger is to replicate updated records to another database.

For more information about replication, see the chapter on data replication in the *Progress Database Administration Guide and Reference*.

Press **PUT** to display the trigger code. Figure 9–4 shows an example of a delete trigger procedure.

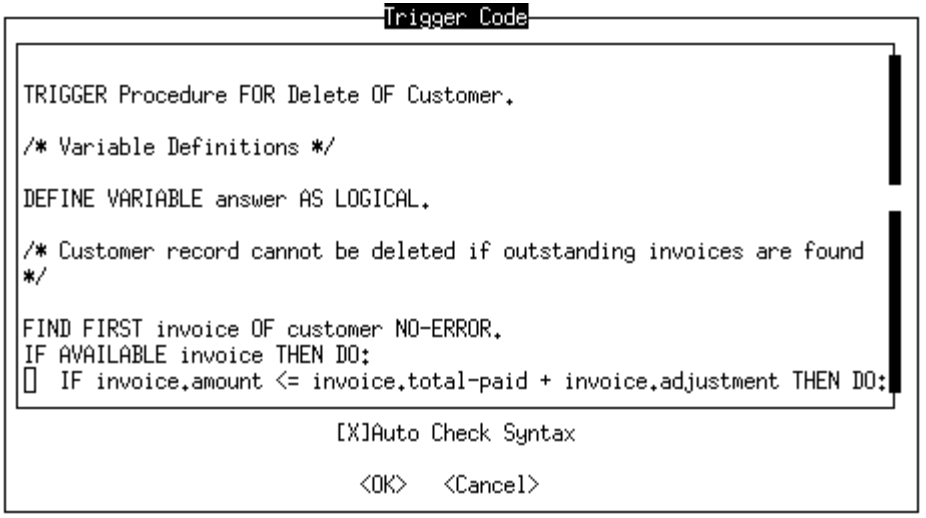


Figure 9–4: Trigger Code Dialog Box

Table String Attributes Option

Choose the String Attributes option to control how much space Progress allocates within an r-code text segment for a string. It is important when you translate a string that you specify enough space for the translated string. String attributes can also specify the text justification within the r-code and whether or not the Translation Manager should translate the string. When you choose the String Attributes button, the Table String Attributes dialog box shown in Figure 9–5 appears.

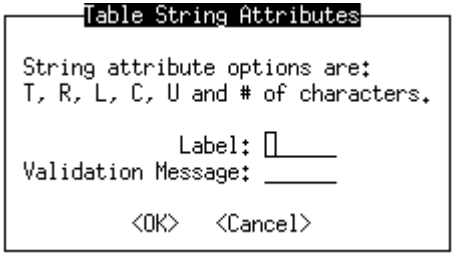


Figure 9–5: Table String Attributes Dialog Box

This dialog box contains the following user-interface elements:

Label

Specifies the string attribute characters for the table label.

Validation Message

Specifies the string attribute characters for the table validation message.

Use the following syntax to enter the string attribute label:

```
[ R | L | C | T ] [ U ] [ max-length ]
```

Table 9–3 describes the string attribute control characters you can enter in the label field.

Table 9–3: String Attribute Characters

Character	Description
R	Specifies to right justify the string.
L	Specifies to left justify the string.
C	Specifies to center the string.
T	Specifies to remove trailing spaces during string display. Progress does not physically remove the spaces from the string, but replaces them with null values. Progress does not display null values.
U	Indicates that the Translation Manager and the compiler should not translate the string.
max-length	Specifies the amount of space allocated for the text segment. You can specify a number from 1 to 999.

You can specify the characters in any order but can only use each character once. Also, you cannot use any of the letters in conjunction with each other, except for U. If you do not specify the maximum length, Progress uses the length of the original string.

If you do not specify justification, the Progress Compiler left-justifies the string by default, unless the string is a column label. If the string is a column label for a decimal or integer column or a side label, it is right justified.

Table 9–4 lists several examples of string attributes.

Table 9–4: String Attribute Examples

Example	Meaning
L15	Specifies a width of 15 characters within the r-code text segment and left-justifies the string within the allocated space.
15R	Specifies a width of 15 characters within the r-code text segment and right-justifies the string within the allocated space.
T20	Specifies a width of 20 characters and trims the leading and trailing spaces within the allocated space.
UC20	Specifies a width of 20 characters and centers the string within the allocated space. Also indicates that the Translation Manager should not translate the string.
20RU	Specifies a width of 20 characters and right-justifies the string within the allocated space. Also indicates that the Translation Manager should not translate the string.
30	Specifies a width of 30 characters.

See the Character-String Literal reference entry in the *Progress Language Reference* for more information about string attributes and string translation.

9.1.2 Schema→ Add New Table

Choose Schema→ Add New Table to create a new table definition. When you choose this option, the window shown in Figure 9–6 appears.

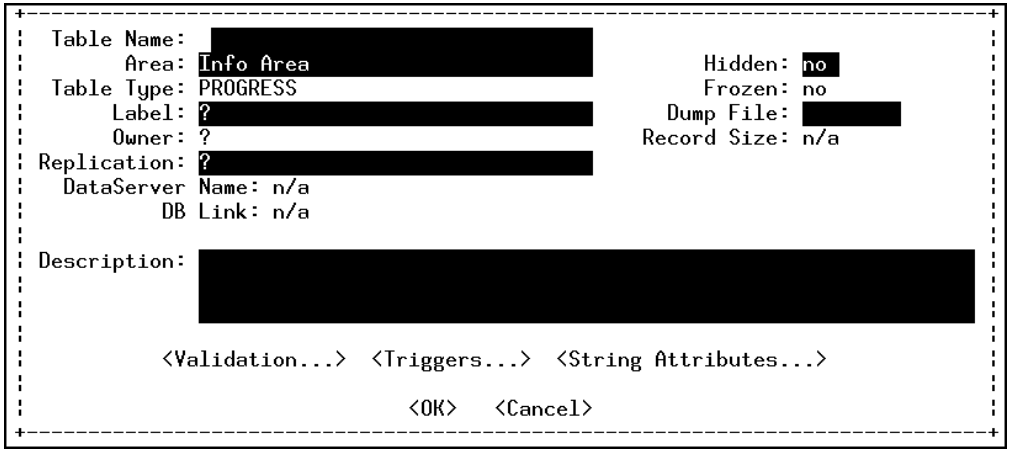


Figure 9–6: Add New Table Window

See the "Schema→ Modify Table" section earlier in this chapter for a description of the user-interface elements for this window.

9.1.3 Schema→ Delete Table(s)

Choose Schema→ Delete Table(s) to delete one or more tables from the working database. Progress prompts you for verification before it deletes the tables you select.

9.1.4 Schema→ Field Editor

Choose Schema→ Field Editor to modify schema definitions for a field. When you choose this option, Progress alphabetically lists all the tables defined for your database. Select the table you want to edit. Progress lists all the fields defined for the selected table. Figure 9–7 shows the fields defined for the customer table.

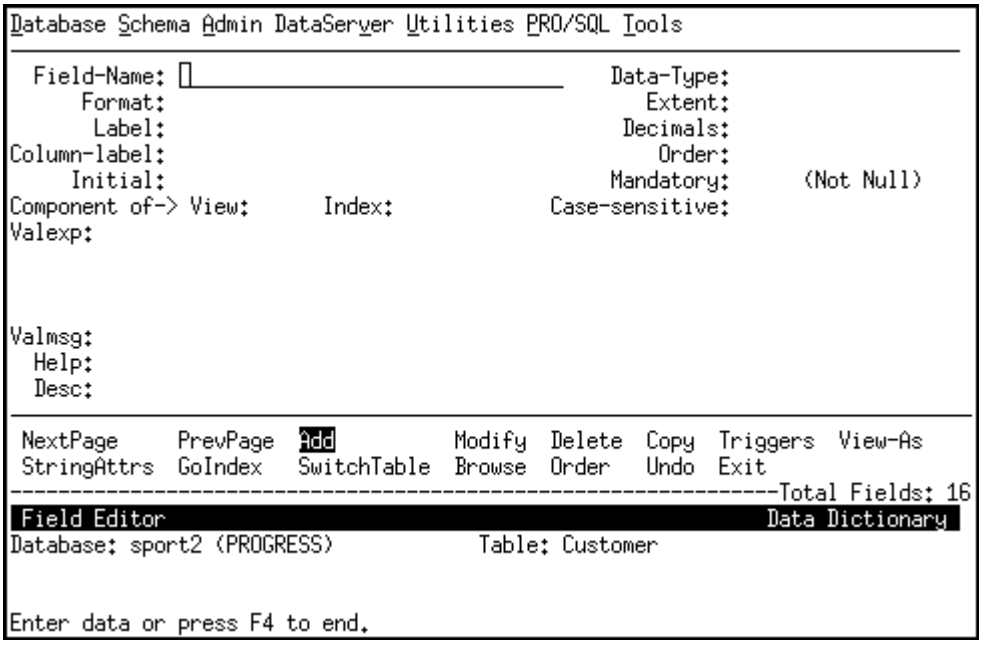


Figure 9–7: Field Editor Window

From this window you can add, edit, delete, or view field attributes defined for the current table. Table 9–5 describes the options for the Field Editor.

Table 9–5: Field Editor Options *(1 of 2)*

Option	Description
NextPage	Displays the next page of fields.
PrevPage	Displays the previous page of fields.
Add	Adds a new field.
Modify	Updates field attributes.
Delete	Deletes a field.
Copy	Copies fields from table definitions.
Triggers	Defines or modifies triggers.
View-As	Defines the 4GL View-As phrase for field widget type.

Table 9–5: Field Editor Options *(2 of 2)*

Option	Description
StringAttrs	Defines or modifies string attributes.
GoIndex	Goes to the Index Editor.
SwitchTable	Switches to a different table.
Browse	Displays the field attributes.
Order	Toggles between displaying fields in alphabetical order or in the order defined by the values in their _Order fields. Progress displays a message at the bottom of the window indicating how the fields are currently ordered.
Undo	Undoes any changes you made to the field attributes for this Field Editor session.
Exit	Saves your changes, exits the Field Editor, and returns to the Data Dictionary Main Menu.

Choose an option, then select the field you want to edit. The following sections describe the Add, Triggers, View-As, and GoIndex options in more detail.

Add Option

Choose the Add option from the Field Editor dialog box to create a new field in the selected table. When you choose this option, a window similar to the one shown in Figure 9–8 appears.

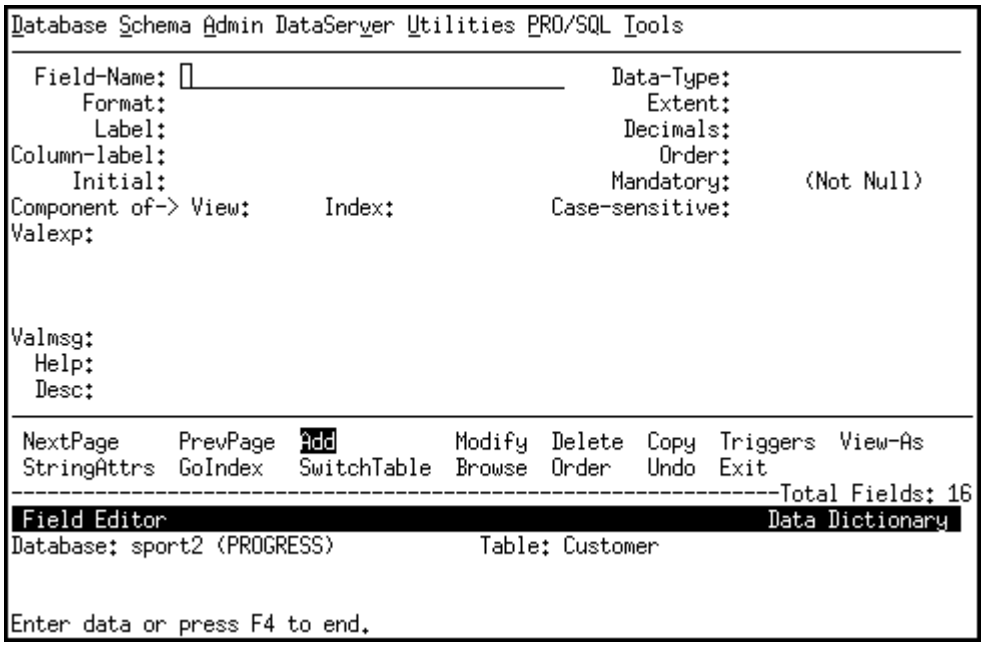


Figure 9–8: Field Add Option Window

NOTE: In previous versions of Progress, adding a field to an existing database table was a time-consuming process for large databases. The DBMS had to update the data in each row of the table to reflect the newly added field. Progress now provides a fast schema change when you are adding fields to a table. Fast schema change is the default behavior.

The Field Add window contains the following user-interface elements:

- **Field-Name**

  Specifies the field name. Names can be up to 32 characters long and can consist of alphabetic characters, digits, and the characters $, &, #, %, -, and _. Field names must begin with A–Z or a–z. You cannot use Progress keywords as field names. Field names are not case sensitive.

  If you select a table name, Progress displays all the field data from the existing field.

- **Format**

  Specifies the length of a field and the way the data is shown on the screen and in printed reports. Progress automatically supplies a default format for each data type, but you can change it. You can also override formats you define in the Data Dictionary by using the

FORMAT option with the Format phrase. For more information about formatting in procedures, see the *Progress Programming Handbook*.

You can also add storage information for some non-Progress databases. See the appropriate Progress DataServer Guide for more information.

The following list describes the default format for each of the data types:

- **Character** — The default format for a character field is x(8). The x represents an alphanumeric character position and the 8 indicates the number of characters to display. Table 9–6 describes the different symbols you can use to define a character format.

Table 9–6: Character Format Symbols

Symbol	Description
x	Represents any character.
n	Represents a digit or a letter. A space is not allowed.
a	Represents a letter. A space is not allowed.
!	Represents a letter that is converted to uppercase during input. A space is not allowed.
9	Represents a digit. A space is not allowed.
(*n*)	Number that indicates how many times to repeat the previous format character. For example, !(5) is the same as !!!!! and represents 5 characters that are to be converted to uppercase when entered.
fillchar	Represents any character or characters you want to fill a display. For example, if you display the value abc with a format of x(3)***, the displayed value is abc***. To use X, N, A, !, and 9 as fill characters, you must precede the character with a tilde (~). To use a left parenthesis (( ) as a fill character after a non-fill character, you must precede it with a tilde (~). See the *Progress Programming Handbook* for more information about fill characters.

Table 9–7 shows examples of the different formats of a character field.

Table 9–7: Character Display Format Examples

Format	Value in Field	Display
xxxxxxxx	These are characters	These ar
x(9)	These are characters	These are
x(20)	These are characters	These are characters
xxx	These are characters	The
AAA–9999	abc1234	abc–1234
(999) 999–9999	6172754500	(617) 275-4500
!!	ma	MA

Progress truncates trailing spaces in character fields. If a character field contains only one space, Progress truncates the space to a null value. You can use the TRIM function to truncate leading and trailing spaces.

- **Integer** and **decimal** — The default display format for an integer field is –>,>>>,>>9. The default display format for a decimal field is –>>.>>9.99. Table 9–8 describes the format characters for numeric display formats.

Table 9–8: Decimal and Integer Format Characters *(1 of 3)*

Character	Description
( )	Progress displays parentheses if the number is negative. If you use one parenthesis (left or right), you must use the other.
string1	A string made up of any characters except plus (+), minus (-), greater than (>), less than (<), comma (,), digits (0–9), letter z (z or Z), asterisk (*), or period (.).

Table 9–8: Decimal and Integer Format Characters *(2 of 3)*

Character	Description
+	Progress replaces this character with a plus sign if the number is positive, and a negative sign if the number is negative. You can also use only one plus or minus sign or CR, DR, or DB, or one set of parentheses in a numeric data format. The position of the plus sign in the number format can also hold a digit if the number is positive.
–	When you use this character to the left of the decimal point, Progress replaces this character with a minus sign if the number is negative and a space or null if the number is positive. When you use this character to the right of the decimal point, Progress replaces this character with a minus sign if the number is negative and a space if the number is positive.
>	Progress replaces this character with a digit if that digit is not a leading zero. If the digit is a leading zero, Progress replaces this character with a null and moves characters to the left, one space to the right, if you are using top labels. If you are using side labels, Progress left-justifies the characters.
,	Progress displays this character as a comma unless it is preceded by a > or Z. If the comma is preceded by a > and the > is replaced by a leading zero, Progress replaces the comma with a null. If the comma is preceded by a Z and the Z is replaced by a space, Progress replaces the comma with a space.
9	Progress replaces this character with a digit, including cases where the digit is a leading zero.
Z	Progress replaces this character with a digit. If the digit is a leading zero, Z suppresses that digit and puts a space in its place.
*	Progress replaces this character with a digit. If the digit is a leading zero, Progress replaces that zero with an asterisk (*).
.	This character represents a decimal point and is displayed as a period (.).

Table 9–8: Decimal and Integer Format Characters *(3 of 3)*

Character	Description
<	Progress uses this character in conjunction with > to implement floating-decimal format. The < character (up to 10) must appear to the right of the decimal and be balanced by an equal or greater number of > characters left of the decimal. Progress displays a digit in a position formatted with < when the corresponding > is a leading zero (and the stored value has the required precision).
DR, CR, DB	Progress displays these characters if the number is negative. If the number is positive, Progress displays spaces in place of these characters. Progress does not treat these characters as sign indicators when you specify *string2*; Progress considers them part of *string2*.
string2	A string made up of any characters except plus (+), minus (-), greater than (>), comma (,), any digit (0–9), letter z (z or Z), or asterisk (*).

When specifying a numeric data format, you must use at least one of the following characters: 9, Z, *, or >.

Table 9–9 shows examples of the different formats for a numeric field.

Table 9–9: Numeric Display Format Examples *(1 of 2)*

Format	Value in Field	Display
9999	123	0123
9,999	1234	1,234
$zzz9	123	$123
$>>>9	123	$123 [1]
$–>,>>9.99	1234	$1,234.00
$>,>>9.99	1234	$1,234.00
#–zzz9.999	–12.34	#–12.340

Table 9–9: Numeric Display Format Examples *(2 of 2)*

Format	Value in Field	Display
Tot=>>9Units	12	Tot=12Units
$>,>>9.99	–12.34	????????? [2]
$>,>>9.99	1234567	????????? [3]
>>,>99.99<<<	12,345.6789	12,345.68[4]
>>,>99.99<<<	1,234.5678	1,234.568[4]
>>,>99.99<<<	123.45	123.45[4]
>>,>9.99<<<	12.45678	12.45678[4]

[1] This display value is right justified if it has a column label, left justified if it has a side label.

[2] There is a negative sign in the value –12.34, but the display format of $>,>>9.99 does not accommodate that sign.

[3] The value 1234567 is too large to fit in the display format of $>,>>9.99.

[4] Floating-decimal display format. The < character must follow the decimal point and be balanced by an equal or greater number of > characters.

NOTE: If you use the European Numeric Format (-E) startup parameter, Progress interprets commas as decimal points and decimal points as commas when displaying or prompting for numeric values. However, always enter formats in the Dictionary as described above. For more information about formatting in procedures, see the *Progress Programming Handbook*.

- **Date** — The default date format is mm/dd/yy. Date formats specify a two-digit month and a two-digit day. You can use a slash (/) or a hyphen (-) as a separator. You can specify a year with two or four digits. When -yy is set at 1950, the default (allows you to set dates for the next century), Progress determines if the two-digit value in the date is greater or less than 50. If the date value is greater than 50, Progress writes the date for the twentieth century. If the date value is less than 50, Progress writes the date for the twenty-first century. For example, if you start Progress with -yy 1950, years 50-99 are treated as 1950-1999, and years 00-49 are treated as 2000-2049. For more information about the Progress date format, see the chapter on startup

parameters in the *Progress Startup Command and Parameter Reference*. Table 9–10 shows some date display format examples.

Table 9–10: Date Display Format Examples

Format	Value in Field	Display
99/99/99	8/10/2001	08/10/01
99/99/9999	08/10/2090	08/10/2090
99-99-99	08/10/1993	08-10-93
99-99-99	08/10/2090	???????? [1]
999999	08/10/1997	031097
999999	08/10/01	081001
99999999	08/10/2001	08102001

[1] The value 8/10/2090 is too large to fit into the display format. If a field's value does not fit in the display format, then Progress displays each integer as a ?.

- **Logical** — The default logical format is YES/NO. You can define any strings to represent those TRUE/FALSE values. If you define your own logical values, the false value cannot begin with "y" or "t", and a true value cannot begin with "n" or "f". If input is coming from a file and you have defined a format for a logical field that is something other than TRUE/FALSE or YES/NO, you can still use TRUE/FALSE or YES/NO as input to that logical field.

Table 9–11 shows logical display format examples.

Table 9–11: Logical Display Format Examples *(1 of 2)*

Format	True	False
yes/no	yes	no
Yes/No	Yes	No

Table 9–11: Logical Display Format Examples *(2 of 2)*

Format	True	False
true/false	true	false
shipped/waiting	shipped	waiting

Label

Specifies the label that represents the field in windows or in printed reports. The default value is a question mark (?). If you leave the question mark and do not supply a label, Progress uses the field names as the label. If you replace the question mark with a space, Progress uses no label for the field. Labels can be up to 30 characters long. Table 9–12 shows some field label examples.

Table 9–12: Field Label Examples

Format	Label You Define	Label Progress Displays
Name	?	Name
Curr-bal	Unpaid Bal	Unpaid Bal
Sales-rep	space	

You can override the labels you define in the Data Dictionary by using LABEL, NO-LABEL, or COLUMN-LABEL options in a Format phrase, or NO-LABEL in a Frame phrase. Progress uses COLUMN-LABEL only for fields that do not have SIDE-LABEL. See the *Progress Programming Handbook* for more information about these phrases.

Column-Label

Specifies that you want to use a different label when the data are listed in columns. If you do not specify a column label, Progress uses the label specified in the Label field. If you do not enter a label, Progress uses the field name. You can override these labels by using NO-LABEL or COLUMN-LABEL options in a Format phrase, or NO-LABEL in a Frame phrase.

If you want the column label to have more than one line (stacked), separate each line of the label with an exclamation point (!). For example, if you want the label for the Curr-bal

field to be Unpaid Balance, with the word “Unpaid” displayed above the word “Balance,” enter Unpaid!Balance for the column label. If you want to use the exclamation point as one of the characters in a column label, you must use two exclamation points (!!). Any spaces to the right or left of the exclamation point become part of the label.

Initial

Defines the initial value for the field. Each data type has an initial value. When you create a new record for a table, each field contains this initial value. You can change a field’s default initial value when you define the field. Table 9–13 lists the default initial values for each data type.

Table 9–13: Field Initial Values

Data Type	Initial Value
Character	Null string (displays spaces)
Integer	0
Decimal	0
Logical	no (false)
Date	? (unknown value; displays spaces)
Raw	Null string (displays spaces)
RECID	? (unknown value; displays spaces)

You can use a question mark (?) as a special character to represent an unknown value. This lets you handle data even when some critical item of information is not yet known. If you put a single question mark in any field, Progress treats the item of data as an unknown value.

You can also use TODAY as the initial value for a date field. When you create a new record, Progress fills in the current date as the initial value for the field.

Component of View

Displays information only; it is not updatable. It tells you whether the field is used in an SQL view. When you use a field as a component of an SQL view, you cannot delete it.

Component of Index

Displays information only; it is not updatable. It tells you whether the field is a component of an index.

Data-Type

Determines the kind of data values the field can store. Table 9–14 describes the available data types.

Table 9–14: Data Types *(1 of 2)*

Data Type	Description
Character	Contains data of any kind. Character data can include uppercase and lowercase characters and can consist of alphabetic characters, digits, and the characters $, &, #, %, –, and _. Although Progress allows character field display formats of up to 255 characters, restrict the format length of a character field to the input/output line width of your terminal (typically 80 characters) by specifying the appropriate length in the Format field.
Integer	Contains only whole numbers. They can be positive or negative, from –2,147,483,648 through 2,147,483,648, inclusive. If you enter spaces as the value of an integer field, Progress stores the value of that field as 0.
Decimal	Contains decimal numbers up to 50 digits. You can use up to 10 digits to the right of the decimal point. If you enter spaces as the value of a Decimal field, Progress stores the value of that field as 0. When you choose decimal as the data type, Progress enables the Decimal field. You specify the number of decimal places in the Decimal field.
Date	Contains dates from 1/1/32768 B.C. to 12/31/32767 A.D., inclusive. You can specify dates in this century with a two-digit year, such as 8/9/93, or a four-digit year, 8/9/1993. Dates in other centuries require a four-digit year. If you enter dates without the year, Progress assumes the current year. If you supply spaces as the value of a date field, Progress stores that field as an unknown value and displays it as spaces.
Logical	Contains logical values such as yes/no or true/false.

Table 9–14: Data Types *(2 of 2)*

Data Type	Description
Raw	Contains the value of a field from a non-Progress database that has a data type with no equivalent in Progress. When you want to work with Raw data, you generally work with the data at the byte level. A Raw value cannot be displayed.
Recid	Contains the record ID.

Extent

Defines the extent of an array field. Most fields represent a single value. However, array fields contain multiple elements. For example, the Mnth-shp field of the time table is an array field. It contains 12 elements, one for every month of the year. The extent is the number of elements contained in an array. If you define a field with an extent greater than 0, that field is an array field.

Decimals

Defines the decimal places for a field. When you define a decimal field, you must define the number of digits to the right of the decimal point. For example, Max-credit has been defined for two digits to the right of the decimal point to accommodate dollars and cents. If the field is not a decimal, you cannot enter a value.

Order

Specifies the default display order of a field. By default, the Data Dictionary numbers fields in the order you enter them, by increments of 10. This defines where Progress lists the field in the field selection list in this table. You can set the order for this field in the table. This display order is not related to the order in which the data are stored in your database. In addition, you can override the Dictionary display order of fields in your procedures by naming the fields in the order you want to display them.

The default order numbering is in increments of 10 to let you insert fields in between. The numbers do not have to be contiguous, nor do they have to follow in even-numbered increments. For example, if you decide to add a field called Category to the customer table, and you want it to appear by default on your window between Cust-num and Name, you might assign 15 as its order value. If you want to change the order number increments to 20, you can do so by choosing the Reorder Fields option of the Schema menu. From the same menu option, you can also change the order values to reflect the alphabetical order of the table names.

Mandatory

Specifies if the field is mandatory. If you define a field as mandatory, it cannot contain an unknown value (?). However, it can have a space as its value. The default value is no. If you accept no for this field, you indicate that the unknown value is an allowed value for the field.

Case-sensitive

Specifies if the field is case sensitive. The default value is no. Because case-sensitive fields depart from the Progress standard, they are not recommended. However, if you require strict adherence to the ANSI SQL standard or if you are using a DataServer that supports case sensitive fields, you might have to define all character fields as case sensitive. Once you define a field as case sensitive, you can redefine it, unless it is a component of an index. If a field is a component of an index, you cannot change its case sensitivity unless the index is undone.

You can index case-sensitive fields and group them with case-insensitive field components in an index. With case-sensitive indexes, JOHN, John, and john are all unique values. However, they do not sort next to each other. All uppercase letters sort ahead of all lowercase letters. Define case-sensitive variables to hold values moving to and from case-sensitive fields.

Valexp

Specifies the validation criteria for the field. You can enter up to 63 characters on each of the four lines. For messages that exceed 252 characters, you can specify an include file enclosed in brackets (use the syntax: {*filename.i*}). There is no default value.

NOTE: You should use field validation for backward compatibility for procedure-driven applications. For event-driven programs, Progress Software Corporation recommends that you use field ASSIGN triggers instead of field validation.

When a user enters a value for a field, you might want to test it to make sure it is a valid entry for the field. The Valexp option lets you define a test or validation expression. The expression must be logical; that is, it must be a valid Progress expression that produces a true or false result.

For example, the validation expression for the Cust-num field in the customer table is cust-num > 0. When the user enters a customer number, Progress validates that number against valexp. If the number is greater than 0, valexp is true, and the validation succeeds. If the number is less than 0, the validation fails. Progress then displays the text in the Valmsg option.

When you write procedures, you can override any validation expressions you define in the Data Dictionary. If you define validation for an array field in the Data Dictionary, the validation only works if you update the entire array. The validation does not work if you update the array elements individually.

Valmsg

Defines the validation message for a field. You can enter up to 63 characters on each line. You cannot define (and do not need) a validation message unless you specify a validation criteria for a field. If the result of Valexp is false (the validation fails), Progress displays the text specified in Valmsg. For example, the validation expression for the Cust-num field in the customer table is cust-num > 0. When the user enters a 0 for the customer number, Progress displays this message:

```
Customer number must be greater than zero.
```

Because Progress treats the message you define as constant (literal) text, it cannot contain references to the number of the fields or variables. If you want to use fields, variables, or expressions in validation messages, use this VALIDATE option in a Frame phrase.

Help

Defines help information for the field. For certain fields, users might be unsure of the kind of data they need to enter. Therefore, you can specify a help message to indicate what information to enter. Progress displays this message whenever users are prompted for input to the field. For example, the State field in the customer table uses this help message:

```
Enter standard state abbreviation.
```

Desc

Describes the field. You might want to document the purpose of a field just as you might supply a description for a table. Progress does not use this option when running procedures; it is strictly to help you document your application.

Field Triggers Option

Choose the Triggers option from the Field Editor to define ASSIGN trigger programs for the field. ASSIGN triggers execute when you assign a value to a database field. When you choose the Triggers option, the Field Triggers dialog box similar to the one shown in Figure 9–9 appears.

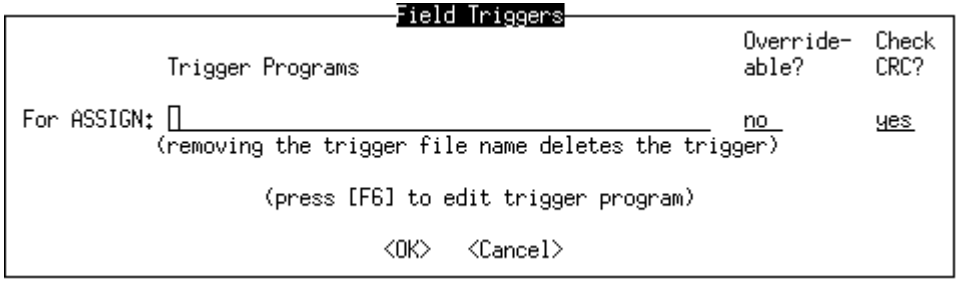


Figure 9–9: Field Triggers Dialog Box

This dialog box contains the following user-interface elements:

For ASSIGN

Specifies the name of the trigger procedure. If you want to create or edit the procedure, press **PUT** and a trigger editor appears.

Overridable?

Specifies whether a session trigger can override the schema trigger. The default value is no. For more information about session and schema triggers, the chapter on database triggers in the *Progress Programming Handbook*.

Check CRC?

Specifies whether Progress verifies the cyclical redundancy check (CRC) code. The default value is no. When Progress checks the code, it compares the stored CRC value of the procedure to the CRC of the trigger when it fires. If they are different, Progress does not run the procedure. This ensures that no one changes the trigger code without proper permission.

You can also edit the trigger program in the Progress Editor.

View As Option

Choose the View As option from the Field Editor to define the View-As phrase for the type of field widget. When you choose this option, Progress displays the View-As Phrase dialog box as shown in Figure 9–10.

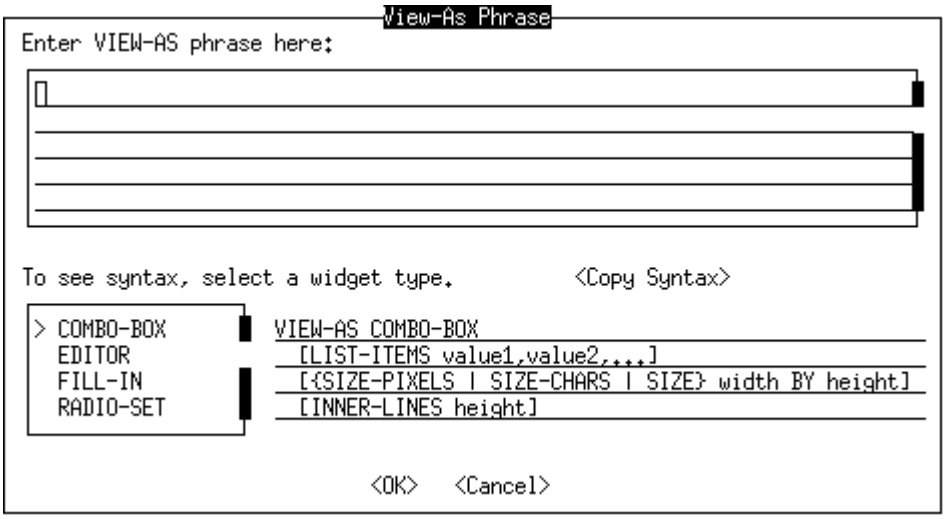


Figure 9–10: View-As Phrase Dialog Box

The list of available widgets depends on the data type of the field. Table 9–15 lists which widgets you can use for each data type. The numeric data type refers to the decimal, integer, and date data types.

Table 9–15: View As Widgets *(1 of 2)*

Widget Type	Data Type		
	Character	Numeric	Logical
Combo-box	√	√	√
Editor	√	–	–
Fill-in	√	√	√
Radio-set	√	√	√

Table 9–15: View As Widgets *(2 of 2)*

Widget Type	Data Type		
	Character	Numeric	Logical
Selection-list	√	–	–
Text	√	√	√

When you select a widget type, Progress displays the appropriate View-As phrase syntax for the field. Choose the Copy Syntax button to copy the syntax to and overwrite any existing text in the editor area of the dialog box. After you edit the View-As phrase, press **GO**. Progress compiles the code. If there are errors in your code, Progress displays the code and lists the errors. If the code is syntactically correct, Progress stores the value and returns you to the Field Editor. When you use this field in an application, it appears as the specified widget type. For more information about defining widgets, see the chapter on widgets and handles in the *Progress Programming Handbook*.

GoIndex Option

Choose the GoIndex option to access the Index Editor. When you choose this option, a window similar to the one shown in Figure 9–11 appears.

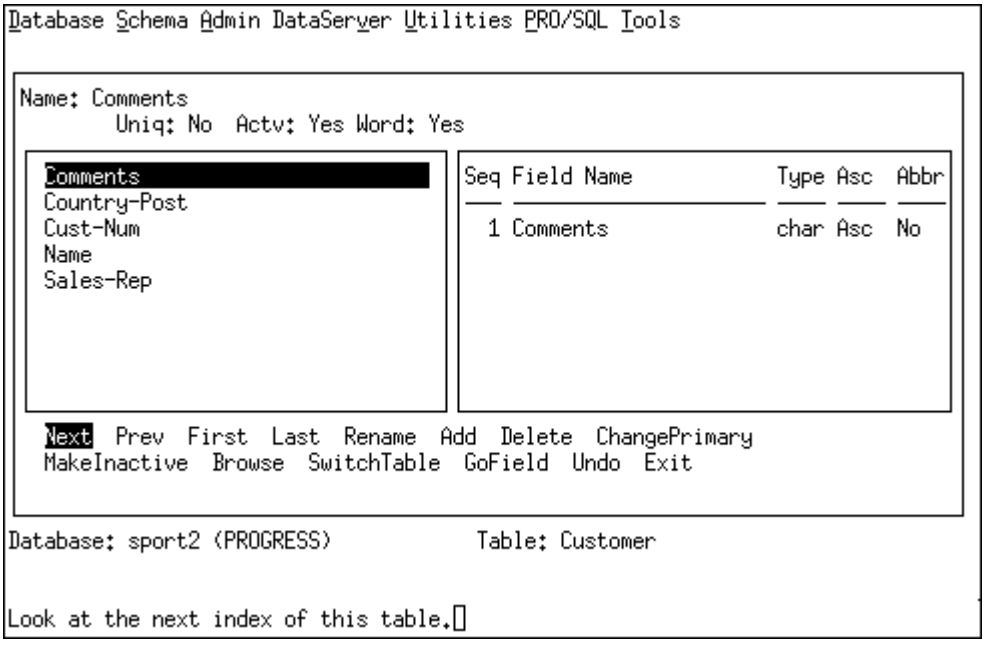


Figure 9–11: GoIndex Option Window

9.1.5 Schema→ Reorder Fields

Choose Schema→ Reorder Fields to resequence the fields alphabetically by field name or numerically by order number. When you choose this option, Progress alphabetically lists all the tables defined for your database. Select the table whose fields you want to reorder. Progress displays the Resequence Order of Fields dialog box similar to the one shown in Figure 9–12.

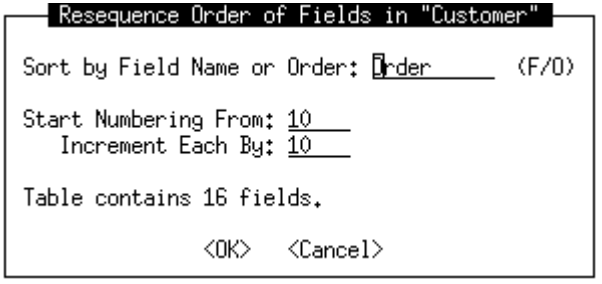


Figure 9–12: Resequence Order of Fields Dialog Box

9.1.6 Schema→ Global Field Name Change

Choose Schema→ Global Field Name Change to change the name of a field in all occurrences in the working database. This option is only available when there is a working database. When you choose this option, Progress displays the Global Field Name Change dialog box shown in Figure 9–13.

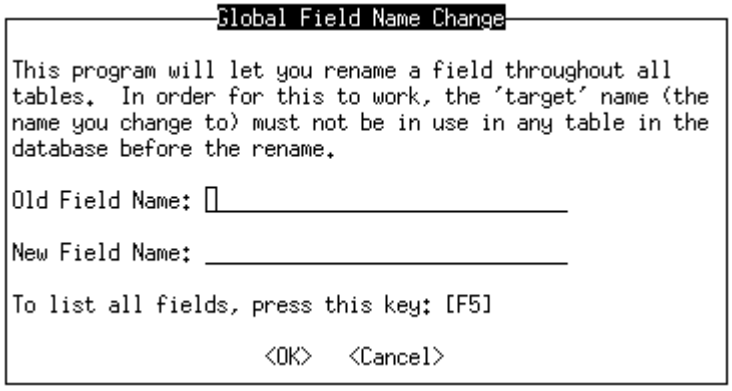


Figure 9–13: Global Field Name Change Dialog Box

Specify the current field name and the new name. When you specify the old field name, Progress checks for a field with the old field name in the working database. If none exists, a message tells you so. If you specify an existing field name, you can enter the new field name.

If you try to change the name of a field to a name that is already in use somewhere in the database, the Data Dictionary displays an error message and does not accept the new field name.

To view a list of all the fields in the database, press **GET**. You can pick a field from the list that appears. To remove the list without picking a field, press **END-ERROR**.

Once you specify acceptable field names in both fields and choose OK, Progress prompts you to verify the change.

9.1.7 Schema→ Index Editor

Choose Schema→ Index Editor to define, rename, or delete indexes for the selected table. An index is a field or combination of fields Progress uses to rapidly retrieve a particular record in a table. A single index can be made up of multiple fields. For example, if you know you are always going to access the order-line table by using a combination of the Order-num and Line-num fields, you might want to create an index with those two fields as its components.

When you choose this option, Progress alphabetically lists all the tables defined for your database. When you select a table, a window similar to the one shown in Figure 9–14 appears.

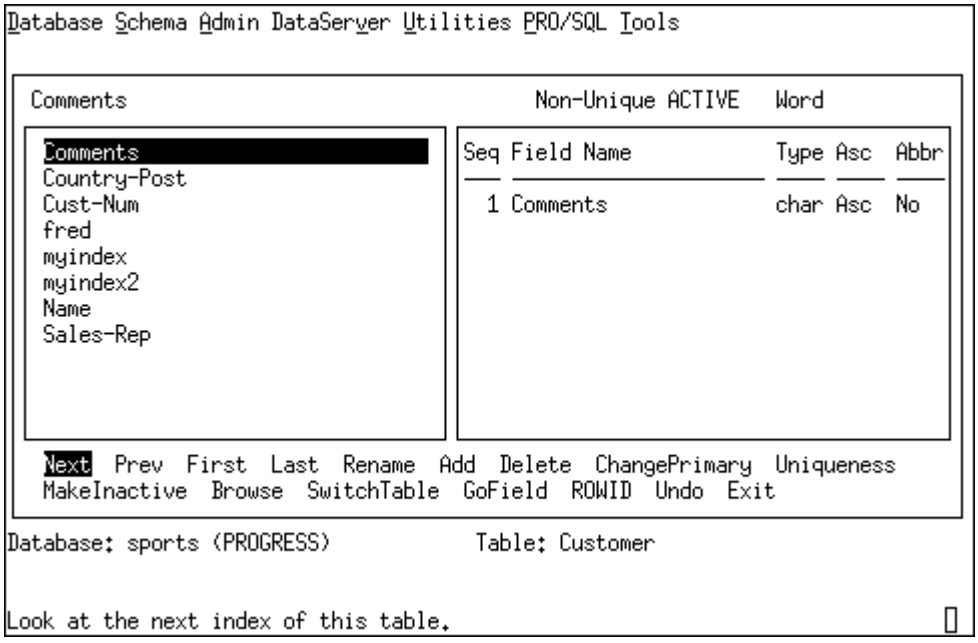


Figure 9–14: Index Editor Window

NOTE: The current table must contain fields before you can define an index.

Use the Index Editor to display the existing indexes and their information for the selected table.

The Index Editor window contains the following fields:

Name

Specifies the index name. It does not have to be the same name as the field name. Index names can be up to 32 character long and can consist of alphabetic characters, digits, and the characters $, &, #, %, –, and _. In addition, index names must begin with a letter (A–Z or a–z). You cannot use reserved Progress keywords as index names. Index names are not case sensitive.

Unique

Defines whether every entry in the index must be different. Enter yes if only one record in a table has a particular index value. For example, the customer table cannot contain two records with the same customer number, so customer number can be a unique index. Conversely, since many customers might be in the same sales region, sales-region should not be a unique index.

ACTIVE

Defines whether the index is initially active, that is, whether it updates itself every time a new record is created, deleted, or modified. If you are defining an index in a very large table, it is more efficient to define the index as initially inactive. (When an index is inactive, you cannot use it to retrieve or order records.) You can subsequently activate the index by running the PROUTIL utility with the IDXBUILD qualifier. This is much faster than building the index from within the Data Dictionary when the database contains a large amount of data. For more information about PROUTIL, see the *Progress Database Administration Guide and Reference*.

ROWID

For non-Progress databases only. See your Progress DataServer Guide for more information on the ROWID property.

Word

Defines whether you can search on any word within a field. A word index contains all the words from a text field or array of text fields; thus, you can search for records that contain specific words or phrases. All words in the field are index entries. For more information about word indexes, see the chapter on database access in the *Progress Programming Handbook*.

Seq

Defines the order in which fields form the index. For example, cust-order is a compound index, formed by the Cust-num and the Order-num fields. The Cust-num field has a sequence number of 1, and the Order-num field has a sequence number of 2.

Field Name

Displays the field name.

Type

Specifies the field data type. See the "Schema→ Field Editor" section in this chapter for more information about data types.

Asc

Specifies that the index sorts records in ascending order.

Abbr

Specifies that you can search an index using the first few characters of a field without using the BEGINS phrase, if the field is a character data field. Indexes not comprised of character data require an exact match. If the index has more than one field, this setting applies only to the last field.

Table 9–16 describes the options for the Index Editor window.

Table 9–16: Index Editor Options *(1 of 2)*

Option	Description
Next	Displays the next index in the table.
Prev	Displays the previous index in the table.
First	Displays the first index in the table.
Last	Displays the last index in the table.
Rename	Renames the selected index. The name you enter cannot already be used to name an index in the current database.
Add	Adds a new index to the table.
Delete	Deletes the selected index. You cannot delete a primary index without making another index the primary index.
ChangePrimary	Changes the primary index for the table.
MakeInactive	Makes an active index inactive. (When an index is inactive, you cannot use it to retrieve or order records.) To reactivate an index, you must run PROUTIL with the IDXBUILD qualifier.
Browse	Browses through the entire index component list. This is especially helpful when the index is made up of multiple fields.

Table 9–16: Index Editor Options *(2 of 2)*

Option	Description
SwitchTable	Shows the list of tables in the database, then switches to another table to edit its indexes.
GoField	Accesses the Field Editor.
Undo	Undoes changes you made to the index attributes during this session.
Exit	Saves your changes, exits the Index Editor, and returns to the Data Dictionary Main Menu.

Choose the field you want to edit, then choose a menu option. The following section describes the Add option in more detail.

Add Option

When you choose the Add option from the Index Editor, Progress prompts you for the name of the new index. When you enter data in the Name, Unique, Active, and Word fields and press **RETURN**, a window similar to the one shown in Figure 9–15 appears.

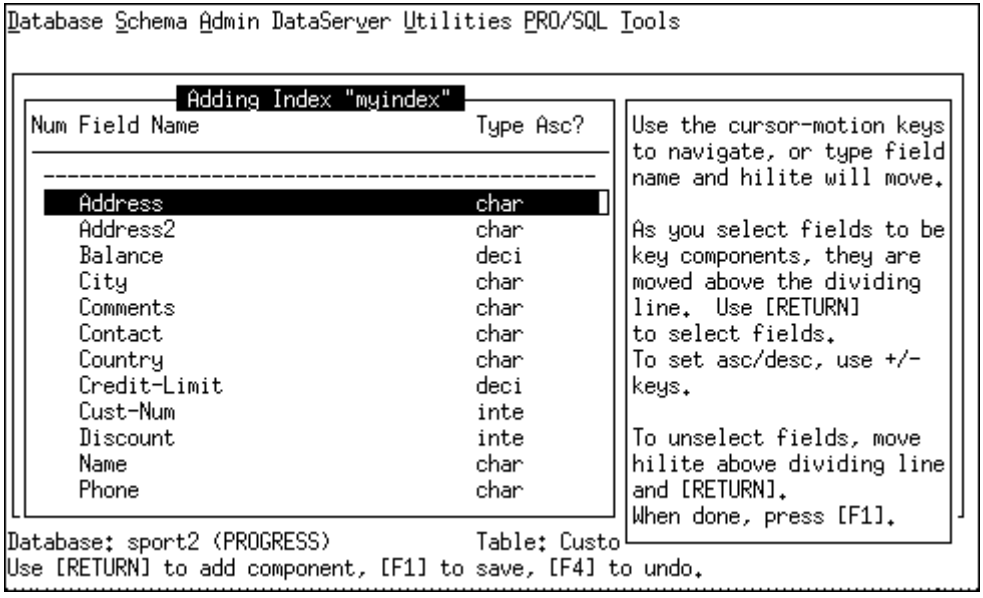


Figure 9–15: Index Add Option Window

Progress lists the available fields and their data types. When you choose a field, Progress prompts you to specify whether you want the index component to be ascending or descending. After you specify the sort order, Progress lets you choose more fields. You can specify up to 10 fields to define an index.

Progress lists the selected field above the divider line on the window. Select all your fields, then press **GO**. Progress prompts you to specify whether you want to use the Abbreviate option for character fields then returns you to the Index Editor and adds the new index to the list.

9.1.8 Schema→ Sequence Editor

Choose Schema→ Sequence Editor to view, define, modify, and delete sequences. *Sequences* are database objects that provide sequential values within any integer range and with your choice of increment. When you choose this option, a window similar to the one shown in Figure 9–16 appears.

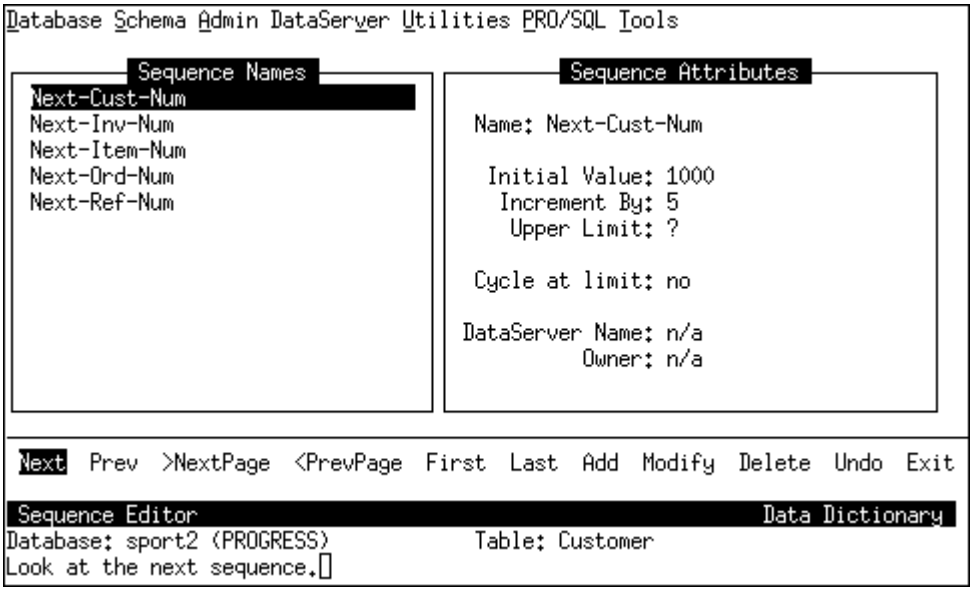


Figure 9–16: Sequence Editor Window

The Sequence Editor displays the existing sequences and their attributes as described below:

Name

Enter a valid Progress identifier as a name for your sequence. The name can be up to 32 characters long.

Initial Value

Enter an initial integer value for the sequence, or accept the default of 0. The initial value becomes the lower limit if the Increment By value is positive and becomes the upper limit if the Increment By value is negative. You cannot specify an initial value that is greater than your upper limit or less than your lower limit, if defined.

Increment By

Enter a positive or negative integer value by which you want to change the sequence value at each increment, or accept the default of 1. If you enter a positive value, then each increment increases the sequence toward a defined upper limit. If you enter a negative value, then each increment decreases the sequence toward a defined lower limit, and the Upper Limit label changes to Lower Limit.

Upper Limit or Lower Limit

Enter an integer value to define the upper or lower limit of the sequence, or accept the default value of unknown (?). If you specify a positive value for the Increment By field, you can only define an upper limit. If you specify a negative value, you can only specify a lower limit. You cannot specify an upper limit that is less than the initial value. Likewise, you cannot specify a lower limit that is greater than the initial value. If you accept the default, Progress uses the maximum integer value on your system as the defined upper limit and the minimum integer value as the defined lower limit.

Cycle at limit

Indicates whether the sequence becomes a cycling or terminating sequence. A cycling sequence restarts at the initial value when it reaches the upper limit or decrements beyond its lower limit, providing non-unique values for the life of the sequence. A terminating sequence stops incrementing when it reaches the upper limit or stops decrementing when it reaches its lower limit, providing unique values as long as you do not explicitly reset it. The default is no, indicating a terminating sequence.

DataServer Name

For non-Progress databases only. Displays the name of the corresponding non-Progress sequence, if there is one.

Owner

For non-Progress databases only. Displays the name of the user who has permission to access the sequence.

Table 9–17 describes the options for the Sequence Editor.

Table 9–17: Sequence Editor Options

Option	Description
Next	Displays the next sequence.
Prev	Displays the previous sequence.
>NextPage	Displays the next set of sequences.
<PrevPage	Displays the previous set of sequences.
First	Displays the first sequence.
Last	Displays the last sequence.
Add	Adds a new sequence.
Modify	Updates sequence attributes.
Delete	Deletes the selected sequence.
Undo	Undoes changes you made to the sequence structures during this session.
Exit	Saves your changes, exits the Sequence Editor, and returns to the Data Dictionary main window.

10

Admin Menu Reference

This chapter describes the Admin menu and dialog boxes. Use the Admin menu to:

- Dump and load database definitions and files
- Implement security
- Create bulk loader description files

10.1 Admin Menu

Table 10–1 describes the menu that appears when you choose this option. More detailed descriptions of the options follow the table.

Table 10–1: Admin Menu

Admin	
Dump Data and Definitions →	Dumps data definitions, table contents, SQL views, user table contents, collation/conversion tables, auto-connect records, sequence definition, and sequence current values.
Load Data and Definitions →	Loads data definitions, table contents, SQL views, auto-connect records, user table contents, and sequence values. Also, reconstructs bad load records.
Security →	Specifies database security.
Export Data →	Exports data files in a specific format to the specified directory.
Import Data →	Imports data in a specific format to the working database.
Create Bulk Loader Description File...	Creates a bulk loader description file.

10.1.1 Admin→ Dump Data and Definitions

Choose this option to dump data and definitions.

NOTE: When a menu option prompts you for a filename only, you can specify a complete pathname.

Table 10–2 describes the menu that appears when you choose this option. More detailed descriptions of the options follow the table.

Table 10–2: Dump Data and Definitions Menu *(1 of 2)*

Dump Data and Definitions	
Data Definitions (.df file)...	Dumps the table definitions.
Table Contents (.d file)...	Dumps table contents.

Table 10–2: Dump Data and Definitions Menu *(2 of 2)*

Dump Data and Definitions	
SQL Views...	Dumps view contents.
User Table Contents...	Dumps user information.
Auto-Connect Records only...	Dumps auto-connect records.
Collation Tables...	Dumps collation or translation tables.
Sequences Definitions...	Dumps sequence definitions.
Sequences Current Values...	Dumps sequence values.
Create incremental .df File...	Compares two Progress database schemas and creates a DF file that contains any differences.

Dump Data and Definitions→ Data Definitions (.df file)

Choose this option to dump table definitions for an individual table or for all the tables in the database. The definition file contains the definitions of tables, which include fields, indexes, and all their characteristics. Progress alphabetically lists all the tables defined for your database. If you choose ALL, the definition file also contains the sequence and auto-connect record definitions.

After you select the tables you want to dump, Progress displays a default name for the file into which it dumps your data definitions (hidden tables are not dumped). This default file is the dump file name you specified when you created the table, or the database name with a .df extension. Progress truncates dump file names to eight characters.

Specify the filename or use the default value. Progress displays each object name as it writes its definition to the definition file. When it has dumped the data definitions, Progress displays a status message and prompts you to continue.

Dump Data and Definitions→ Table Contents (.d file)

Choose this option to dump table contents. Progress alphabetically lists all the tables defined for your database. When you select only one table, Progress displays the Dump Data Contents dialog box for that table. If you select multiple tables, Progress displays the Dump Data Contents for Some Tables dialog box as shown in Figure 10–1.

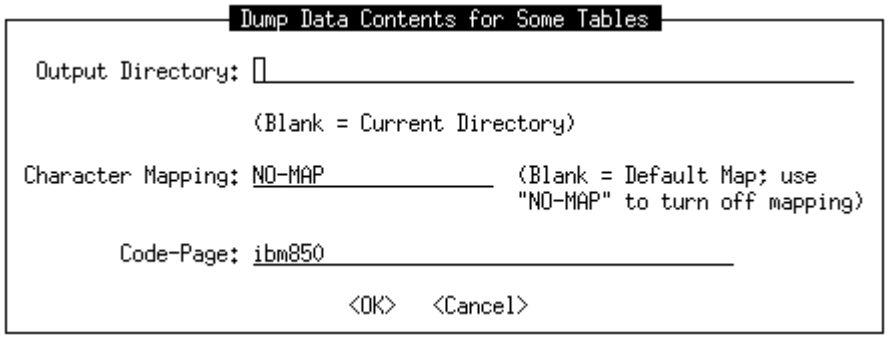


Figure 10–1: Dump Data Contents for Some Tables Dialog Box

This dialog box contains the following user-interface elements:

Output Directory or Output File

Specifies the output directory if you choose to dump multiple tables. Specifies the output file if you choose to dump only one table. The default filename for each table is the dump filename you specified when you created the table, or *table-name*.d.

Character Mapping

Specifies the type of character mapping.

- **Default Map** — Specifies the Progress default mapping.
- **Map** — Specifies the character mapping. Enter the type of mapping in the fill-in field. For example, MAP laserwriter/french or MAP hp2/spanish/italian.
- **NO MAP** — Specifies not to perform any character mapping.

Code-Page

Specifies the default code page (character set). A code page maps characters to numbers (code-points) and defines the character's visual representation (glyph). Code pages usually represent a particular language's alphabet, such as the ibm861 code page (Icelandic).

Progress displays each table name as it writes the table contents to the file's contents file. Progress displays a status message and prompts you to continue.

Dump Data and Definitions→ SQL Views

Choose this option to dump SQL view contents. Progress prompts you for the filename to which it writes the view. The default filename is `_view.d`. Specify the filename. Progress displays a status message and prompts you to continue.

Dump Data and Definitions→ User Table Contents

Choose this option to dump user information. Progress prompts you for the filename to which it writes the user table contents. The default filename is `_user.d`. Specify the filename. Progress displays a status message and prompts you to continue.

Dump Data and Definitions→ Auto-Connect Records Only

Choose this option to dump auto-connect records. Progress prompts you for the filename to which it writes the auto-connect records. The default filename is `_auto.df`. Specify the filename. Progress displays a status message and prompts you to continue.

NOTE: The Data Dictionary does not have a separate option for loading auto-connect records. When you choose Load→ Data Definitions from the menu bar, Progress automatically loads all the table definitions, regardless of the table type.

Dump Data and Definitions→ Collation Tables

Choose this option to dump collation tables, case conversion tables, and translation tables. Progress prompts you for the filename to which it writes the tables. The default filename is `_tran.df`. Specify the filename. Progress displays a status message and prompts you to continue.

NOTE: The Data Dictionary does not have a separate option for loading the collation, case conversion, or translation tables. When you choose Load→ Data Definitions from the menu bar, Progress automatically loads all the table definitions, regardless of the table type.

Dump Data and Definitions→ Sequences Definitions

Choose this option to dump sequence definitions. Progress prompts you for the name of the file to which it writes the sequence definitions. The default filename is `_seqdefs.df`. Specify the filename. Progress displays a status message and prompts you to continue.

NOTE: The Data Dictionary does not have a separate option for loading sequence definitions. When you choose Load→ Data Definitions from the menu bar, Progress automatically loads all the table definitions, regardless of the table type.

Dump Data and Definitions→ Sequences Current Values

Choose this option to dump sequence values. Progress prompts you for the filename to which it writes the sequence values. The default filename is `_seqvals.d`. Specify the filename. Progress displays a status message and prompts you to continue.

NOTE: The Table Contents (.d file) option only dumps table contents. You must dump the sequence values separately.

Dump Data and Definitions→ Create Incremental .df File

Use this option to compare two Progress databases' schemas and create a DF file that contains any differences. You can then use the new DF file to upgrade from a database to the current database.

NOTE: You need at least two databases connected to create an incremental DF file.

When you choose this option, the Create Incremental Definitions File window shown in Figure 10–2 appears.

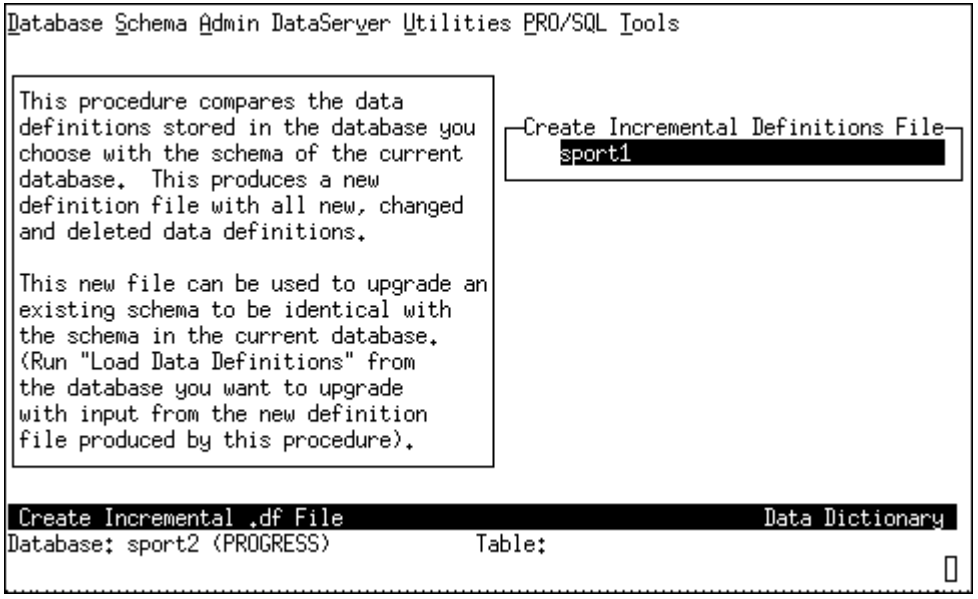


Figure 10–2: Create Incremental .df File Window

Progress lists all connected databases except the working database (which should be the database that has the newest version of the database schema). If you have more than two other databases connected, select the database that has the older version of the database schema. Progress then prompts you for the filename to which you want to write the differences. The

default filename is `delta.df`. Progress displays the file, field, and index names as it compares the databases.

You can then use this file to apply schema changes to an existing database.

NOTE: If you use this option to create a DF file in conjunction with an object file to update schema changes, you must load the DF file and recompile before you can run the new r-code. You must recompile because Progress reorders the indexes during the dump and load procedure.

10.1.2 Admin→ Load Data and Definitions

Choose this option to load table definitions for specified tables or for all the tables in the database.

NOTE: When a menu option prompts you for a filename only, you can specify a complete pathname.

Table 10–3 describes the menu that appears when you choose this option. More detailed descriptions of the options follow the table.

Table 10–3: Load Data and Definitions Menu

Load Data and Definitions	
Data Definitions (.df file)...	Loads table, auto-connect record, sequence, and collation table definitions.
Table Contents (.d file)...	Loads table contents.
SQL Views...	Loads SQL view contents.
User Table Contents...	Loads user table contents.
Sequences Current Values...	Loads sequence values.
Reconstruct Bad Load Records...	Uses the error file and the original data file to build a new data file for bad records.

Load Data and Definitions→ Data Definitions (.df file)

Choose this option to load table and sequence definitions for selected tables. The DF file contains the definitions of tables, which can include fields, indexes, sequences, auto-connect records, collate/translate tables, and all their characteristics. Progress prompts you for the filename that contains the data definitions you want to load into your current working database. The default file is the working physical database name with a .df extension.

You can specify whether you want to stop the load as soon as Progress encounters a bad definition statement. Progress displays each item as it loads the definitions for that object. Progress displays a status message and prompts you to continue. Progress truncates table names to eight characters.

NOTE: When you choose Load Data Definitions→ Data Definitions from the menu bar, Progress automatically loads all the table definitions, including any collation or translation tables. If you change the collate/translate tables, you must rebuild your indexes.

Load Data and Definitions→ Table Contents (.d file)

Choose this option to load table contents. Progress alphabetically lists all the tables defined for your database. After you select the tables whose contents you want to load, Progress prompts you for the name of the contents file or, if there is more than one, the name of the directory that contains the contents files.

If you load table contents for a single table, Progress displays a default name for the file from which you can load your table contents. This default file is always the name of the table dump file, usually the table name with a .d extension.

When you load the contents of more than one table, Progress prompts you for a directory from which it loads the contents files. If you do not specify a directory, Progress loads the files from the current directory. Progress loads each table from the corresponding *table-name*.d file.

Progress also prompts you for an acceptable error rate. As Progress loads records from any files you designate, it might encounter data that cannot be loaded. For example, a record might have data in one of its fields that is too long for that field's defined format. Progress does not load the record. If you specify an error rate of 10 percent, Progress must successfully load 90 records from every set of 100 records loaded. If the error rate exceeds the specified rate, Progress terminates the load. Progress places bad records in the table-name.e file.

If the definition file does not specify a code page, Progress prompts you for one.

NOTE: You can move and reorganize data by performing a binary dump and load. The binary method improves the dump and load performance significantly because the data remains in binary format during the process. For more information about performing a binary dump and load, see Chapter 1, "Database Tasks."

Load Data and Definitions→ SQL Views

Choose this option to load SQL view contents. Progress prompts you for the filename from which it reads the view contents. The default filename is _view.d. Specify the filename. Progress displays a status message and prompts you to continue.

Load Data and Definitions→ User Table Contents

Choose this option to load user information. Progress prompts you for the filename from which it reads the user table contents. The default filename is `_user.d`. Specify the filename. Progress displays a status message and prompts you to continue.

Load Data and Definitions→ Sequences Current Values

Choose this option to load sequence values. Progress prompts you for the filename from which it reads the sequence values. The default filename is `_seqvals.d`. Specify the filename. Progress displays a status message and prompts you to continue.

Load Data and Definitions→ Reconstruct Bad Load Records

Choose this option to build a new data file for bad records from the error file and original data file. If Progress encounters errors while loading a data file, it creates the error file. When you choose this option, the dialog box in Figure 10–3 appears.

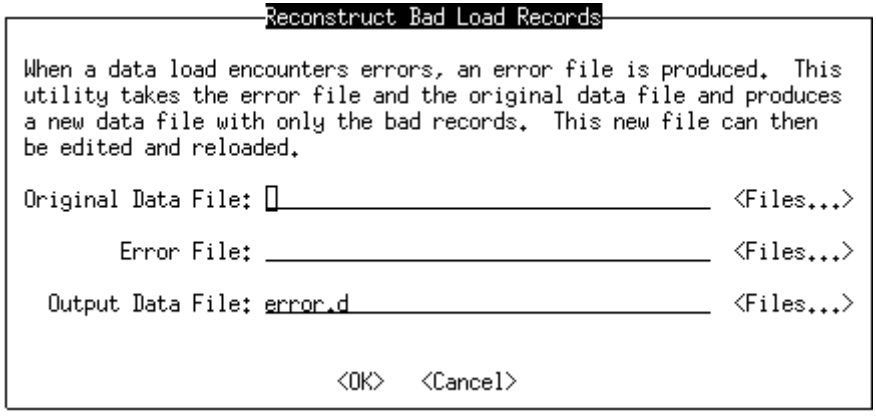


Figure 10–3: Reconstruct Bad Load Records Dialog Box

This dialog box contains the following user-interface elements:

Original Data File

Specifies the name of the file that contains the original data.

Error File

Specifies the name of the error file.

Output Data File

Specifies the name of the new data file. The default filename is `error.d`.

Progress builds the new file. You can use a text editor to edit the new data file and fix the bad records after Progress builds the new file. Once you have fixed the records, you can reload the file.

10.1.3 Admin→ Security

Choose the Security option to perform the following tasks:

- Define access permissions for the tables and fields in your database
- Set up a security administrator
- Define and maintain user IDs and passwords

Table 10–4 describes the menu that appears when you choose this option. More detailed descriptions of the options follow the table. See also the *Progress Database Administration Guide and Reference* and the *Progress Client Deployment Guide* for more information on security administration.

Table 10–4: Security Menu

Security	
Edit User List...	Modifies the user list.
Change Your Password...	Changes your password.
Edit Data Security...	Defines access to security tables and fields.
Security Administrators...	Designates security administrators.
Disallow Blank Userid Access...	Denies blank user IDs access to the working database.
User Report...	Displays or print user information.

Security→ Edit User List

Choose Security→ Edit User List to add or delete users from the user list. You can also change the user name. When you choose this option, the Edit User List window shown in Figure 10–4 appears.

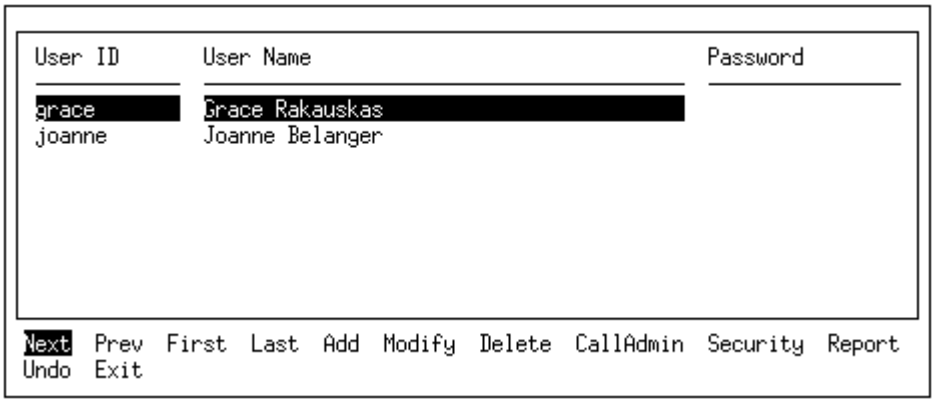


Figure 10–4: Edit User List Window

This window contains the following user-interface elements:

User ID

Lists the users defined for the working database.

User Name

Displays the user name of the selected user.

Password

Indicates whether the selected user has been assigned a password.

Table 10–5 describes the options at the bottom of the Edit User List window.

Table 10–5: Edit User List Window Options *(1 of 2)*

Option	Action
Next	Highlights the next user in the user list.
Prev	Highlights the previous user in the user list.
First	Highlights the first user in the user list.
Last	Highlights the last user in the user list.
Add	Adds a new user.

Table 10–5: Edit User List Window Options *(2 of 2)*

Option	Action
Modify	Modifies the selected user's name.
Delete	Deletes the selected user ID.
CallAdmin	Accesses the Security Administrators dialog box.
Security	Accesses the Data Security option dialog box.
Report	Generates a user report for the working database.
Undo	Discards any changes you made and exits to the Data Dictionary window.
Exit	Exits, saves changes, and returns to the Data Dictionary window.

Once you designate security administrators, only they can edit the user list.

Security→ Change Your Password

Choose this option to change your password. Progress prompts you for your new password. Enter your new password, then choose OK. Remember that passwords are case sensitive. Progress prompts you to verify the new password.

Progress allows all users to change their own passwords.

Security→ Edit Data Security

Choose Security→ Edit Data Security to define compile-time security for an application database. Progress alphabetically lists the names of the tables defined for the working database. When you select a table, Progress displays the current permissions lists for the table. Figure 10–5 shows the default permissions for the customer table.

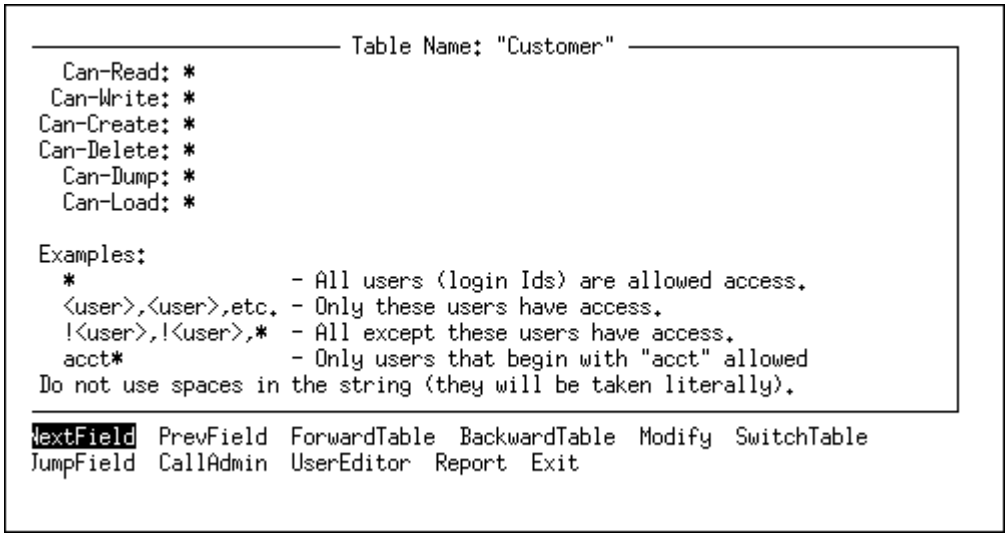


Figure 10–5: Edit Data Security Dialog Box

The upper section of the dialog box displays the six levels of security permissions for the table. By default, Progress initializes all levels with an asterisk (*), which means that all users are allowed to modify table permissions. The lower section describes the syntax for the permissions lists.

The six levels of security permissions for tables are:

Can-Read

Specifies the users who have permission to read a table. The default is an asterisk (all users).

Can-Write

Specifies the users who can write to a table or update records. The default is an asterisk (all users).

Can-Create

Specifies the users who can create new records. In order to create a new record, a user must have Can-Write privileges. The default is an asterisk (all users).

Can-Delete

Specifies the users who can delete records from a table. The default is an asterisk (all users).

Can-Dump

Specifies whether you can dump database or table definitions and data with triggers disabled. The default is an asterisk (all users).

Can-Load

Specifies whether you can load database or table definitions and data with triggers disabled. The default is an asterisk (all users).

Table 10–6 describes the values you use to define the permissions for a table.

Table 10–6: Access Restrictions for Tables

Expression	Meaning
*	All users are allowed access.
user	Only this user has access.
! *user*, *	All users have access except this user.[1]
acct*	User IDs that begin with *acct* have access.

[1] Do not place the asterisk (*) before !user. If you do, all users are allowed access.

Table 10–7 describes the options at the bottom of the Edit Data Security dialog box.

Table 10–7: Edit Data Security Dialog Box Options *(1 of 2)*

Option	Action
NextField	Displays the next field defined for the specified table.
PrevField	Displays the previous field defined for the specified table.
ForwardFile	Displays the permissions for the next table defined for the working database.
Modify	Modifies the permissions for a table or field.
SwitchFile	Displays the list of tables again so you can select a new table.
JumpField	Displays a list of all the fields defined for the current table.

Table 10–7: Edit Data Security Dialog Box Options *(2 of 2)*

Option	Action
CallAdmin	Accesses the Security Administrators dialog box.
Security	Accesses the Data Security option dialog box.
Report	Generates a user report for the working database.
Exit	Exits, saves changes, and returns to the Data Dictionary window.

Once you designate security administrators, only they can use this option.

Security→ Security Administrators

Choose Security→ Security Administrators to designate a security administrator. The Security Administrators dialog box shown in Figure 10–6 appears.

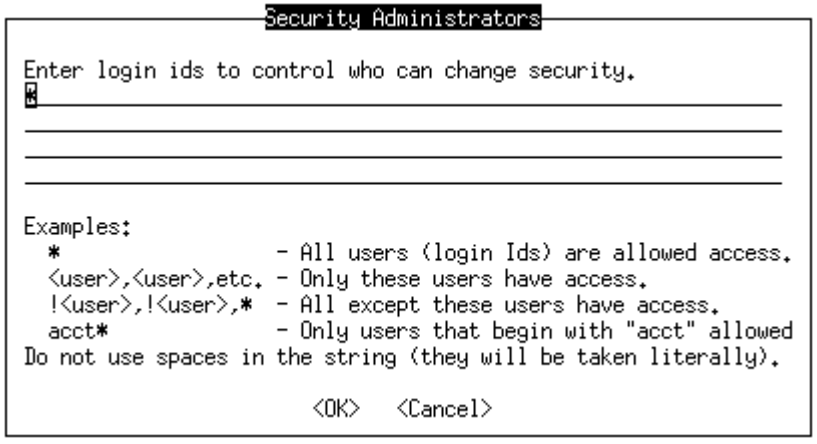


Figure 10–6: Security Administrators Dialog Box

You can enter many user IDs here, but you must include your own user ID. Separate the user IDs with commas, but no spaces. When you are done entering the user IDs, choose OK. Progress prompts you to verify the entries you made.

Once you designate security administrators, only they can use this option.

Security→ Disallow Blank User Access

Choose this option to prevent users who are not listed in the user list from accessing the working database. When you choose this option, Progress displays an alert box prompting you to verify that you want to prevent blank user ID access to the database.

Security→ User Report

Choose Security→ User Report to display or print a list of user IDs and user names, as well as to determine which user IDs are protected by passwords. When you choose this option, Progress displays the user ID list, as shown in Figure 10–7.

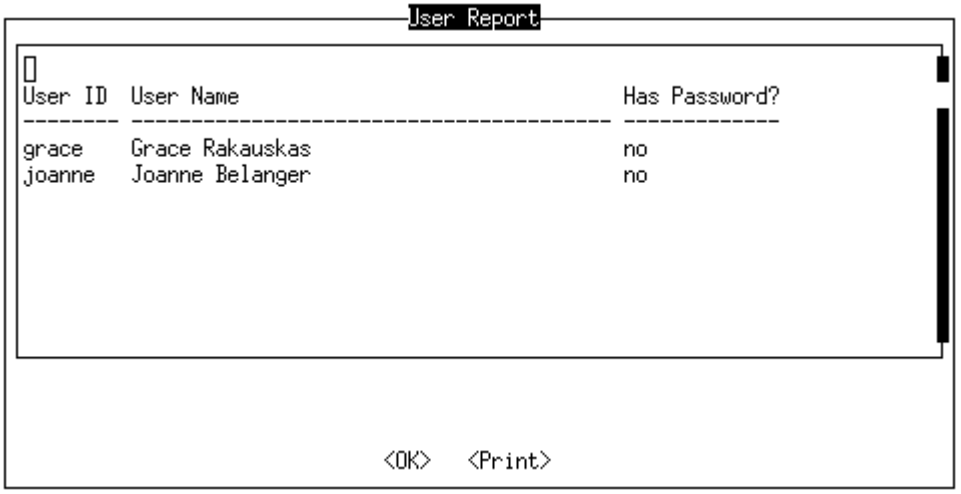


Figure 10–7: User Report Dialog Box

To send the report to a file or printer, choose Print. Progress prompts you for the printer or filename, whether you want to append the report to an existing file, and the page length.

10.1.4 Admin→ Export Data

Choose this option to export data in different formats. Table 10–8 describes the menu that appears when you choose this option. More detailed descriptions of the options follow the table.

Table 10–8: Export Data Menu *(1 of 2)*

Export Data	
DIF ...	Exports data records in DIF format.
SYLK ...	Exports data records in SYLK format.

Table 10–8: Export Data Menu *(2 of 2)*

Export Data	
Text ...	Exports data records in plain text format.
WordStar Merge Data...	Exports data records to WordStar in text format.
Microsoft Word Merge Data...	Exports data records to Microsoft Word in text format.
WordPerfect Merge Data...	Exports data records to WordPerfect in text format.

Choose these options to export data in the specified format. When you choose the desired format, Progress alphabetically lists all the tables defined for your database. When you select a table, a dialog box similar to the one shown in Figure 10–8 opens.

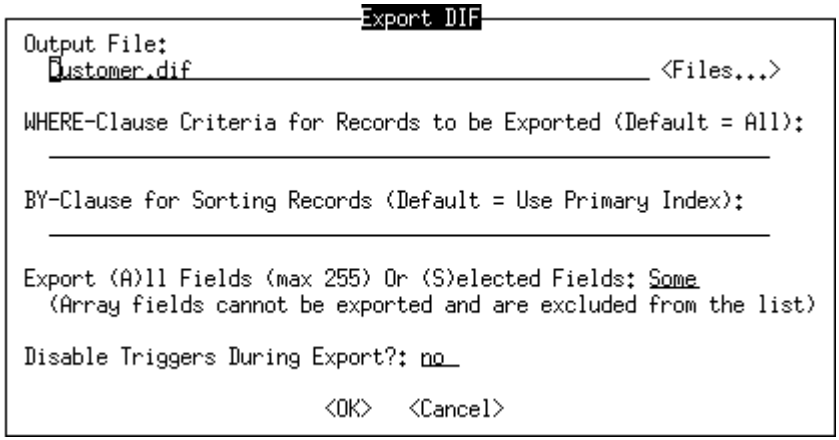


Figure 10–8: Export DIF Dialog Box

This dialog box contains the following user-interface elements:

Output File

Specifies the name of the destination file. Table 10–9 lists the export format file extensions.

Table 10–9: Export Format File Extensions

File Format	Extension
DIF	.dif
SYLK	.sl
Text	.txt
WordStar	.txt
Microsoft Word	.doc
WordPerfect	.txt

WHERE-Clause

Specifies the record selection criteria (for example, WHERE cost > 100). A record becomes a row in a worksheet, and the worksheet starting point is always row 1, column 1. The default is to select all the records.

BY-Clause

Specifies that you want to sort the records using a BY clause (for example, BY name). The default is to use the primary index.

Export Fields

Specifies the fields you want to export. The default value is Some. Each field corresponds to a worksheet column.

If you specify Selected Fields, Progress lists all the fields in the specified file. To select a field, press **SPACEBAR** to highlight the field, than press **RETURN**. Select the fields in the order you want to export them into your destination file. To skip a column in the worksheet, do not press **RETURN** while the field is highlighted. Press **GO** to export the fields.

Disable Triggers During Export?

Specifies whether to disable triggers during the export.

Text Only: After you enter the data into the Export Data dialog box, the Output Record Formatting dialog box shown in Figure 10–9 appears.

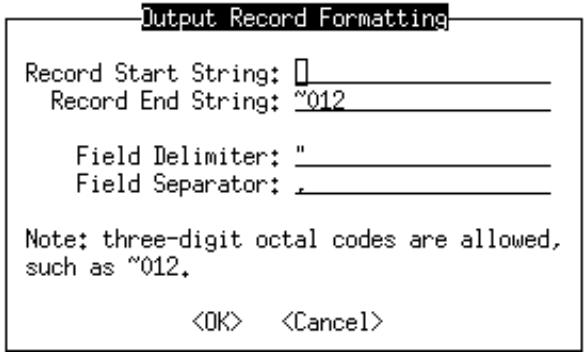


Figure 10–9: Output Record Formatting Dialog Box

This dialog box contains the following user-interface elements:

Record Starter String

Specifies the character that indicates the beginning of a record. Enter the appropriate three-digit code.

Record End String

Specifies the character that indicates the end of a record. The default is a carriage return (~012).

Field Delimiter

Specifies the character that surrounds a field. The default is a quotation mark (").

Field Separator

Specifies the character that separates fields. The default is a comma (,).

NOTE: Do not use a Progress escape character—tilde (~) or backslash (\)—as a field delimiter or field separator.

For all formats, Progress informs you of the number of records it dumped in specified format and prompts you to continue.

10.1.5 Admin→ Import Data

Choose this option to import data with different file formats to the working database. Table 10–10 describes the menu that appears when you choose this option. More detailed descriptions of the options follow the table.

Table 10–10: Import Data Menu

Import Data	
DIF ...	Imports files with DIF format.
SYLK ...	Imports files with SYLK format.
Delimited Text ...	Imports files with text format.
Fixed-Length ...	Imports files in fixed-length file format.
dBASE Definitions ...	Imports dBASE II, III, III+, and IV file structures.
dBASE File Contents ...	Imports files with dBASE format.

Choose Import Data then either the DIF, SYLK, Delimited Text, or dBASE File Contents option to import data in the specified format. When you choose the desired format, Progress alphabetically lists all the tables defined for your database. When you select a table, a dialog box similar to the one shown in Figure 10–10 appears.



Figure 10–10: Import DIF Dialog Box

This dialog box contains the following user-interface elements:

Import File

Specifies the load file.

Import

Specifies the fields you want to import. The default value is Some. Each field corresponds to a worksheet column. Each row in the worksheet becomes a record in the table. For each row, Progress loads worksheet values column by column into the specified record fields.

If you specify Selected Fields, Progress lists all the fields in the specified file. To select a field, press **SPACEBAR** to highlight the field, than press **RETURN**. Select the fields in the order you want to import them into your file. To skip a column in the worksheet, do not press **RETURN** while the field is highlighted. Press **GO** to import the fields.

Disable Triggers During Import?

Specifies whether to disable triggers during the import.

Text Only: To import a file in delimited text format, you must also specify the field separation character.

dBASE File Contents Only: To import dBASE file contents, you must also specify if the file originated on an IBM PC or compatible.

Import Data→ Fixed-Length

Choose Import Data→ Fixed-Length to import data in fixed-length format. When you choose this option, Progress alphabetically lists all the tables defined for your database. When you select a table, the Import Fixed-Length dialog box shown in Figure 10–11 appears.

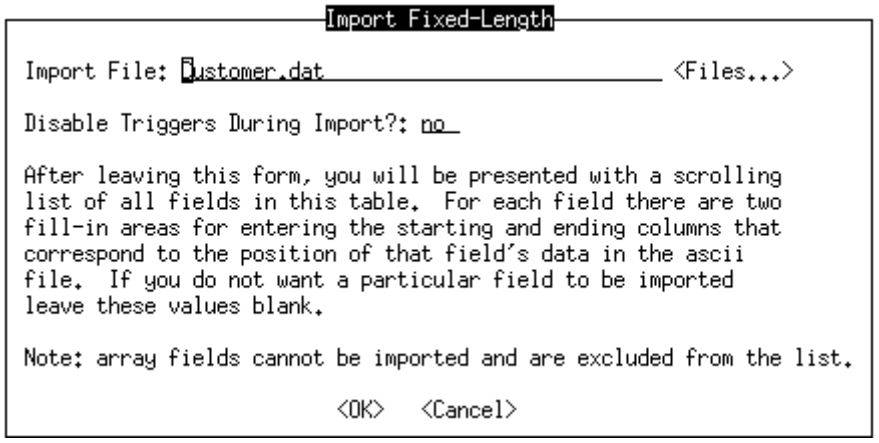


Figure 10–11: Import Fixed-Length Dialog Box

This dialog box contains the following user-interface elements:

Import File

Specifies the load file.

Disable Triggers During Import?

Specifies whether to disable triggers during the import.

When you choose OK, the Fixed Length Field Columns dialog box shown in Figure 10–12 appears.

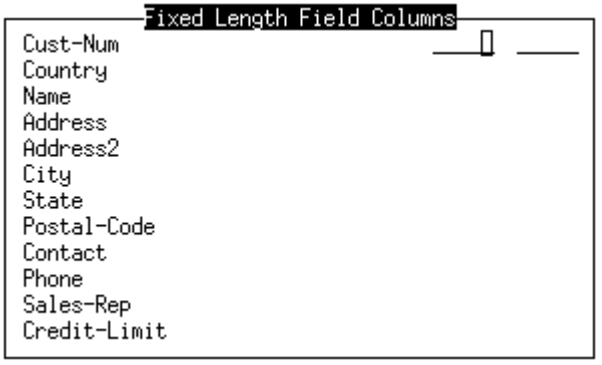


Figure 10–12: Fixed Length Field Columns Dialog Box

This dialog box lists the fields you can import. Select the desired fields.

Import Data→ dBASE Definitions

Choose Import Data→ dBASE Definitions to import dBASE II, III+, and IV file structures. When you choose this option, the dialog box shown in Figure 10–13 appears.

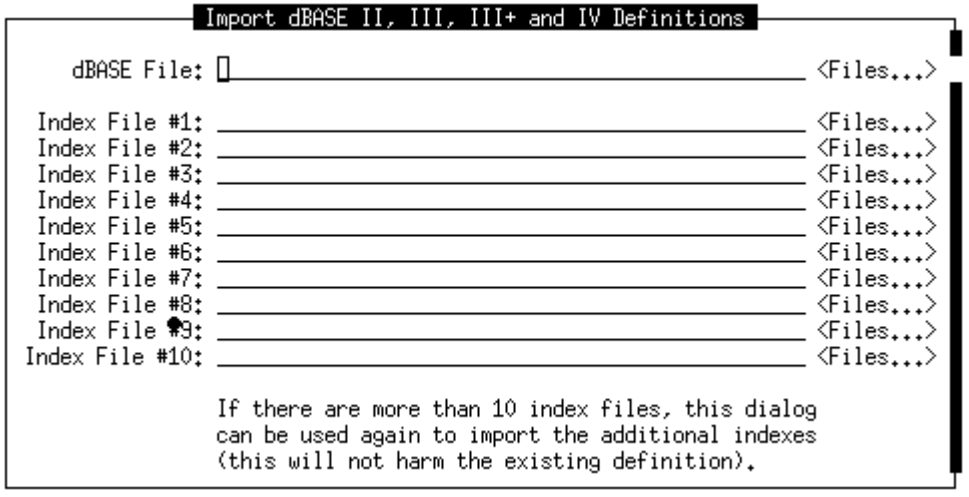


Figure 10–13: Import dBASE II, III, III+ and IV Definitions Dialog Box

Specify the DBF filename and up to 10 index (NDX) names. If you want to import more than 10 indexes, you must rerun the option and specify the remaining indexes.

11

Advanced Menus Reference

This chapter describes the Utilities menu and dialog boxes. Use the Utilities menu to:

- Edit the parameter files
- Freeze and unfreeze database tables
- Interface to the Quoter utility
- Generate include files for use with the 4GL
- Edit auto-connect lists
- Deactivate indexes
- View information about the database and the current session parameters

Also, this chapter describes the PRO/SQL menu and dialog boxes. Use this menu to:

- Manage views created with Progress/SQL statements.

11.1 DataServer Menu

Choose the DataServer Menu to manage your dataservers. Table 11–1 describes the menu that appears when you choose this option. For more information on these menu options, see your DataServer documentation.

Table 11–1: DataServer Menu

DataServer	
DB2/400 Utilities→	Accesses utilities for managing Version 7 DB2/400 DataServers
ODBC Utilities→	Accesses utilities for managing ODBC DataServers.
ORACLE Utilities→	Accesses utilities for managing ORACLE DataServers.

11.2 Utilities Menu

Choose the Utilities menu to edit the parameter files, freeze and unfreeze database tables, interface to the Quoter utility, generate include files for use with the 4GL, edit auto-connect lists, deactivate indexes, and view information about the database and the current session parameters.

Table 11–2 describes the menu that appears when you choose this option. More detailed descriptions of the options follow the table.

Table 11–2: Utilities Menu *(1 of 2)*

Utilities	
Editor for Parameter Files...	Edits a parameter file.
Quoter Functions→	Formats a data file so that Progress can read the data into a Progress database.
Generate Include Files→	Generates an include file.
Edit Progress Auto-Connect List...	Prepares a list of secondary Progress databases to which Progress connects automatically as required when you run a procedure from a primary database.
Freeze/Unfreeze...	Freezes or unfreezes a table.

Table 11–2: Utilities Menu *(2 of 2)*

Utilities	
Index Deactivation...	Deactivates indexes.
Information...	Displays general information about the database and the Progress product you are using.

11.2.1 Utilities→ Editor for Parameter Files

Choose Utilities→ Editor for Parameter Files to view a complete list of startup parameters, fill in the desired values, and write out a PF file (rather than typing in the parameter file with an ordinary text editor). When you choose this option, Progress prompts you for the name of the PF file to edit or create. After you specify the filename, the Edit Parameter File (`param.pf`) dialog box shown in Figure 11–1 appears.

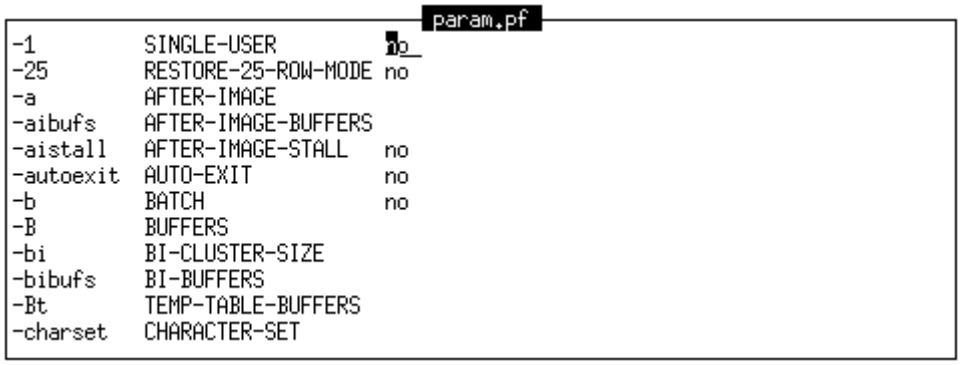


Figure 11–1: Edit Parameter File Dialog Box

Specify the parameters you want, then press **GO**. Progress displays a dialog box similar to the one shown in Figure 11–2. Enter any comments, then choose OK.

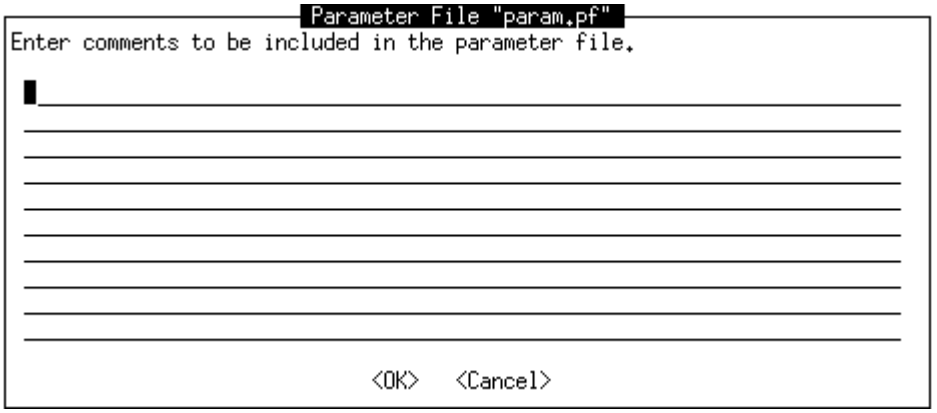


Figure 11–2: Parameter File Comments Dialog Box

11.2.2 Utilities→ Quoter Functions

Choose this option to format a data file so that Progress can read the data into a Progress database. For more information about the quoter utility, see the *Progress Programming Handbook*.

Table 11–3 describes the menu that appears when you choose this option. More detailed descriptions of the options follow the table.

Table 11–3: Quoter Functions Menu

Quoter Functions	
Entire Lines...	Copies your input file to a new file formatted with quotes around each line.
By Delimiter...	Copies your input file to a new file formatted with quotes around each delimited field.
By Column Ranges...	Copies your input file to a new file formatted with quotes around each column-defined field.
Quoter Include File...	Creates an include file that contains a quoter command.

Quoter Functions→ Entire Lines

Choose Quoter Functions→ Entire Lines to copy your input file to a new file formatted with quotes around each line, ready for Progress input. When you choose this option, the Quote Entire Lines dialog box shown in Figure 11–3 appears.

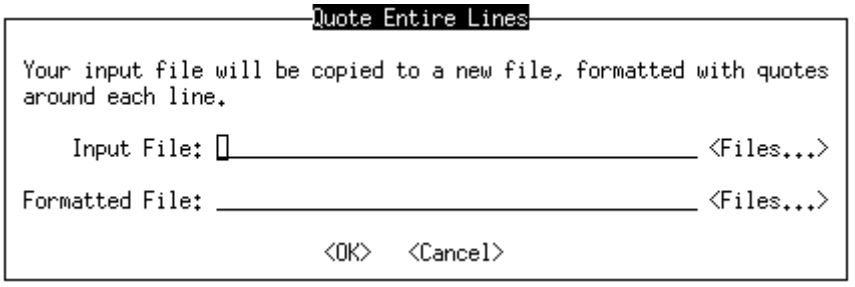


Figure 11–3: Quote Entire Lines Dialog Box

This dialog box contains the following user-interface elements:

Input File

Specifies the filename of the input file to format.

Formatted File

Specifies the filename of the new Progress formatted file.

Files

Displays a dialog box that lets you select a file from a list of existing files.

Quoter Functions→ By Delimiter

Choose Quoter Functions→ By Delimiter to copy your input file to a new file formatted with quotes around each delimited field, ready for Progress input. When you choose this option, the Quote By Delimiter dialog box shown in Figure 11–4 appears.

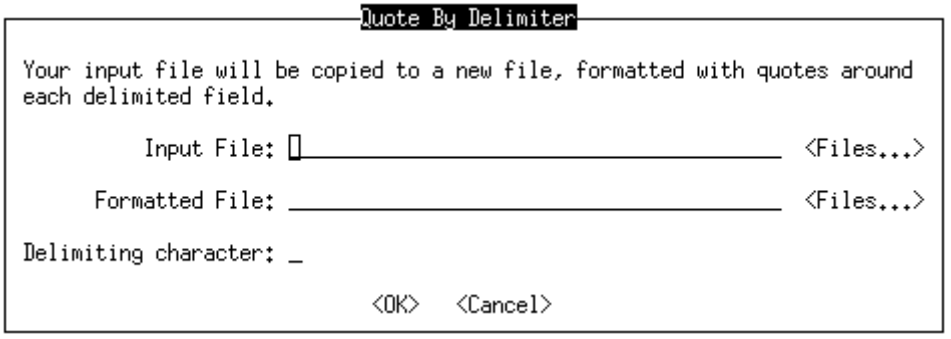


Figure 11–4: Quote By Delimiter Dialog Box

This dialog box contains the following user-interface elements:

Input File

Specifies the filename of the input file to format.

Formatted File

Specifies the filename of the new Progress formatted file.

Delimiting Character

Specifies the character with which to delimit each field.

Files

Displays a dialog box that lets you select a file from a list of existing files.

Quoter Functions→ By Column Ranges

Choose Quoter Functions→ By Column Ranges to copy your input file to a new file formatted with quotes around each column-defined field, ready for Progress input. When you choose this option, the Quote By Column Ranges dialog box shown in Figure 11–5 appears.

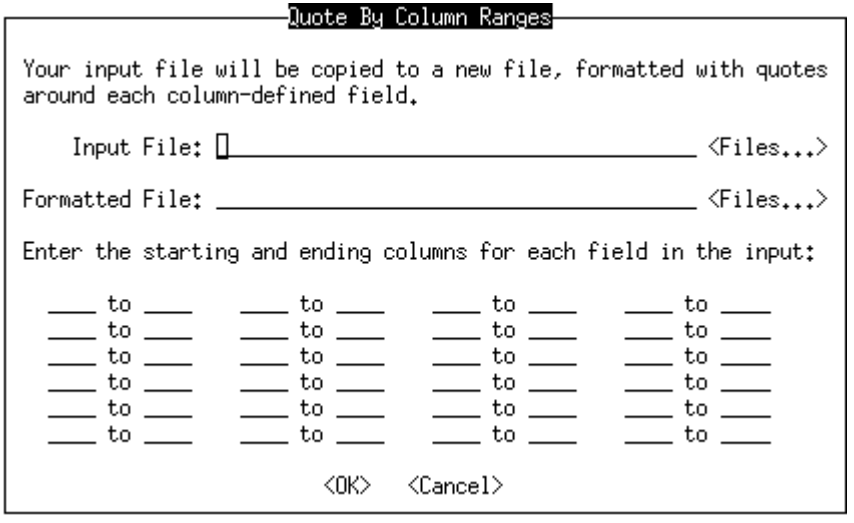


Figure 11–5: Quote By Column Ranges Dialog Box

This dialog box contains the following user-interface elements:

Input File

Specifies the filename of the input file to format.

Formatted File

Specifies the filename of the new Progress formatted file.

Files

Displays a dialog box that lets you select a file from a list of existing files.

Specify the column ranges for each field.

11.2.3 Utilities→ Generate Include Files

Choose this option to quickly prepare an include file to store Progress language statements. Because the ASSIGN, FORM, and DEFINE WORK-TABLE statements can be long and complicated, you might want to store them in an include file. Also, particular instances of these statements are often repeated in different procedures, and using an include file lets you easily insert them into any Progress procedure. See the following reference entries in the *Progress Language Reference* for more information: { }Include File, ASSIGN statement, FORM statement, and DEFINE WORK-TABLE statement.

Table 11–4 describes the menu that appears when you choose this option. More detailed descriptions of the options follow the table.

Table 11–4: Generate Include Files Menu

Generate Include Files	
ASSIGN Statement ...	Generates an ASSIGN statement to copy a table.
FORM Statement ...	Generates a FORM statement for a table.
DEFINE WORK-TABLE Statement ...	Generates a WORK-TABLE statement for a table.

Generate Include Files→ ASSIGN Statement

Use this option to generate an ASSIGN statement to copy a table. Progress alphabetically lists all the tables defined for your database. When you choose a table, Progress prompts you for the filename of the output file. The default filename is *table-name*.i. Progress creates an output file

that contains an ASSIGN statement to copy each field of the table from one buffer record to another and prompts you to continue.

Figure 11–6 shows an example of the ASSIGN statement generated for the customer table in the sports database.

```
/* 08/03/94 COPY assignment */

ASSIGN
  {1}.Cust-Num      = {2}.Cust-Num
  {1}.Country       = {2}.Country
  {1}.Name          = {2}.Name
  {1}.Address       = {2}.Address
  {1}.Address2      = {2}.Address2
  {1}.City          = {2}.City
  {1}.State         = {2}.State
  {1}.Postal-Code   = {2}.Postal-Code
  {1}.Contact       = {2}.Contact
  {1}.Phone         = {2}.Phone
  {1}.Sales-Rep     = {2}.Sales-Rep
  {1}.Credit-Limit  = {2}.Credit-Limit
  {1}.Balance       = {2}.Balance
  {1}.Terms         = {2}.Terms
  {1}.Discount      = {2}.Discount
  {1}.Comments      = {2}.Comments.
```

Figure 11–6: ASSIGN Statement Example

Generate Include Files→ FORM Statement

Choose Generate Include Files→ FORM Statement to generate a FORM statement for a table. Progress alphabetically lists all the tables defined for your database. When you select a table, the Generate FORM dialog box shown in Figure 11–7 appears.

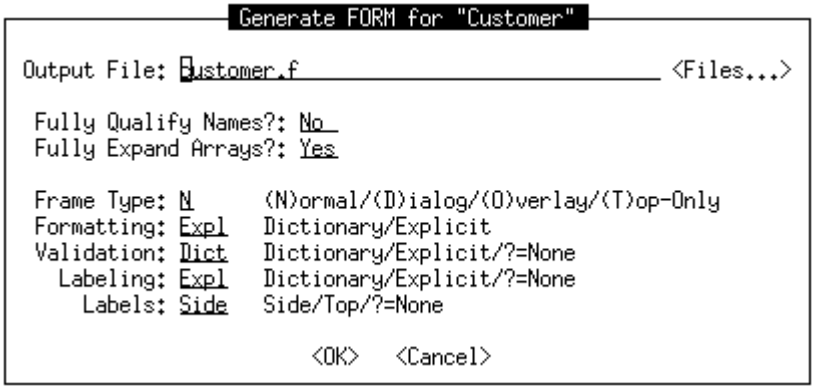


Figure 11–7: Generate FORM Dialog Box

This dialog box contains the following user-interface elements:

Output File

Specifies the filename of the output file. The default filename is the table name with a .f extension.

Fully Qualify Names

Specifies whether to qualify field names with table names (that is, *table-name.field-name*). The default value is No.

Fully Expand Arrays

Specifies whether to fully expand arrays. For example, X[1] is expanded and X is not expanded. The default value is Yes.

Frame Type

Specifies the type of frame to use. The options are Normal, Dialog, Overlay, or Top-Only. The default value is Normal.

Formatting

Specifies the formatting to use. The options are Dictionary or Explicit. Dictionary specifies to use the Data Dictionary default formats without listing them in the FORM statement. Explicit specifies to explicitly list the format definitions in the FORM statement. The default value is Explicit.

Validation

Specifies what type of validation to use. The options are Dictionary, Explicit, or None. The default value is Dictionary. Dictionary specifies to use the Data Dictionary default validation without listing them in the FORM statement. Explicit specifies to explicitly list the validation definitions in the FORM statement. None specifies that you do not want to use validation from the Dictionary nor from FORM statements.

Labeling

Specifies what type of labeling to use. The options are Dictionary, Explicit, or None. The default value is Explicit. Dictionary specifies to use the Data Dictionary default labels without listing them in the FORM statement. Explicit specifies to explicitly list the label definitions in the FORM statement. None specifies that you do not want to any use labels, that is Progress adds NO-LABELS to the WITH phrase of the FORM statements.

Labels

Specifies the label position. The options are Side, Top, or None. The default value is Side.

Figure 11–8 shows an example of the FORM statement generated for the customer table in the sports database.

```
/* 08/03/94 FORM LIKE table Customer */

FORM
  Cust-Num FORMAT ">>>>9"
  Country FORMAT "x(20)"
  Name FORMAT "x(20)"
  Address FORMAT "x(20)"
  Address2 FORMAT "x(20)"
  City FORMAT "x(12)"
  State FORMAT "x(20)"
  Postal-Code FORMAT "x(10)"
  Contact FORMAT "x(20)"
  Phone FORMAT "x(20)"
  Sales-Rep FORMAT "X(4)"
  Credit-Limit FORMAT "->,>>>,>>9"
  Balance FORMAT "->,>>>,>>9.99"
  Terms FORMAT "x(20)"
  Discount FORMAT ">>9%"
  Comments FORMAT "x(60)"
  WITH FRAME customer SIDE-LABELS.
```

Figure 11–8: FORM Statement Example

Generate Include Files→ DEFINE WORK-TABLE Statement

Use this option to generate a DEFINE WORK-TABLE statement for a table. Progress alphabetically lists all the tables defined for your database. When you choose a table, Progress prompts you for the filename of the output file. The default filename is *table-name*`.i`. Progress displays a status message and prompts you to continue.

Figure 11–9 shows an example of the DEFINE WORK-TABLE statement generated for the customer table in the sports database.

```
/* 08/03/94 workfile definition for table Customer */
/* {1} = "", "NEW" or "NEW SHARED" */
/* {2} = "" or "NO-UNDO" */

DEFINE {1} WORK-TABLE wcustomer {2} /* LIKE Customer */
  FIELD Address      AS CHARACTER FORMAT "x(20)"
  FIELD Address2     AS CHARACTER FORMAT "x(20)"
  FIELD Balance      AS DECIMAL   DECIMALS 2 FORMAT "->,>>>,>>9.99"
  FIELD City         AS CHARACTER FORMAT "x(12)"
  FIELD Comments     AS CHARACTER FORMAT "x(60)"
  FIELD Contact      AS CHARACTER FORMAT "x(20)"
  FIELD Country      AS CHARACTER FORMAT "x(20)" INITIAL "USA"
  FIELD Credit-Limit AS DECIMAL   DECIMALS 2 FORMAT "->,>>>,>>9" INITIAL 1500
  FIELD Cust-Num     AS INTEGER   FORMAT ">>>>9"
  FIELD Discount     AS INTEGER   FORMAT ">>9%"
  FIELD Name         AS CHARACTER FORMAT "x(20)"
  FIELD Phone        AS CHARACTER FORMAT "x(20)"
  FIELD Postal-Code  AS CHARACTER FORMAT "x(10)"
  FIELD Sales-Rep    AS CHARACTER FORMAT "X(4)"
  FIELD State        AS CHARACTER FORMAT "x(20)"
  FIELD Terms        AS CHARACTER FORMAT "x(20)" INITIAL "Net30".
```

Figure 11–9: DEFINE WORK-TABLE Statement Example

11.2.4 Utilities→ Edit Progress Auto-Connect List

Choose Utilities→ Edit Progress Auto-Connect List to create or modify the auto-connect list. The *auto-connect list* is a list of secondary Progress databases (with connection parameters) that Progress connects automatically as required during program execution on a primary database. You can define one for each Progress database.

NOTE: To use the Auto-Connect feature for non-Progress databases, use the Edit Connection Parameters option under the appropriate DataServer submenu option. See the appropriate Progress DataServer manual for information on auto-connecting to non-Progress databases.

The Progress auto-connect feature uses information stored in a primary application database to connect to a secondary application database. The primary application database contains the database connection information for the secondary database. When data from the second database is referenced in a compiled application at run time, the primary application database must already be connected before Progress can execute the auto-connect for the secondary application database. Progress executes an auto-connect immediately prior to running a procedure that accesses a database on the auto-connect list. If you connect a database with the CONNECT statement, and that database also has an auto-connect entry in an already connected database, the CONNECT statement uses the information from the auto-connect list; however, the explicit connection information in the CONNECT statement takes precedence. For more information about auto-connect, see the chapter on database access in the *Progress Programming Handbook*.

When you choose this option, the Edit Auto-Connect List window shown in Figure 11–10 appears.

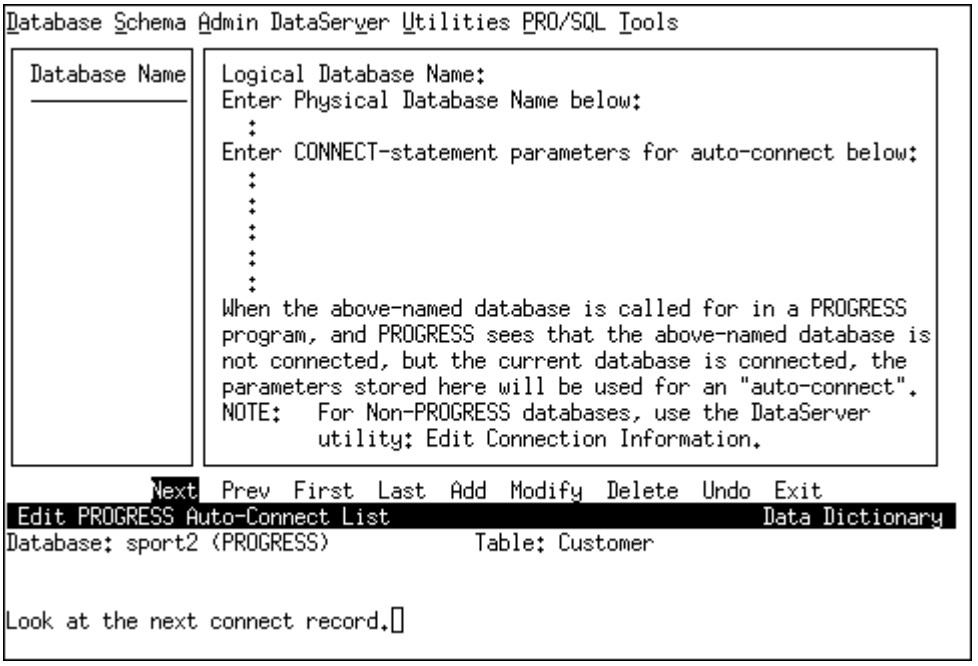


Figure 11–10: Edit Progress Auto-Connect List Window

This window contains the following user-interface elements:

Database Name

Displays the list of databases to which Progress connects automatically.

Logical Database Name

Specifies the logical name of the database highlighted in the Database Name list.

Physical Database Name

Specifies the physical name of the database highlighted in the Database Name list.

CONNECT-Statement Parameters

Specifies any other startup parameters for the database that are not included in the parameter file. Progress Software Corporation recommends that you specify all parameters in the parameter file.

Table 11–5 describes the buttons on the Edit Auto-Connect List dialog box.

Table 11–5: Edit Auto-Connect List Options

Option	Action
Next	Displays the next connect record.
Prev	Displays the previous connect record.
First	Displays the first connect record.
Last	Displays the last connect record.
Add	Adds a new connect record.
Modify	Modifies the selected connect record.
Delete	Deletes the selected connect record.
Undo	Exits, undoes all changes you made in this session to the Auto Connect list, and returns to the Data Administration window.
Exit	Exits, saves changes, and returns to the Data Administration window.

11.2.5 Utilities→ Freeze/Unfreeze Table

Choose Utilities→ Freeze/Unfreeze to freeze or unfreeze a database table. Freezing a database table prevents users from changing the table, field, or index definitions for the table. Progress alphabetically lists all the tables defined for your database. When you select a table, the Freeze/Unfreeze Table dialog box shown in Figure 11–11 appears.

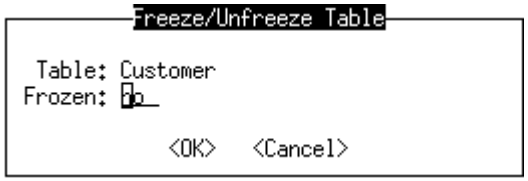


Figure 11–11: Freeze/Unfreeze Table Dialog Box

Specify whether you want Progress to freeze or unfreeze the table.

11.2.6 Utilities→ Index Deactivation

Choose Utilities→ Index Deactivation to deactivate indexes. When you choose this option, the Index Deactivation dialog box shown in Figure 11–12 appears.

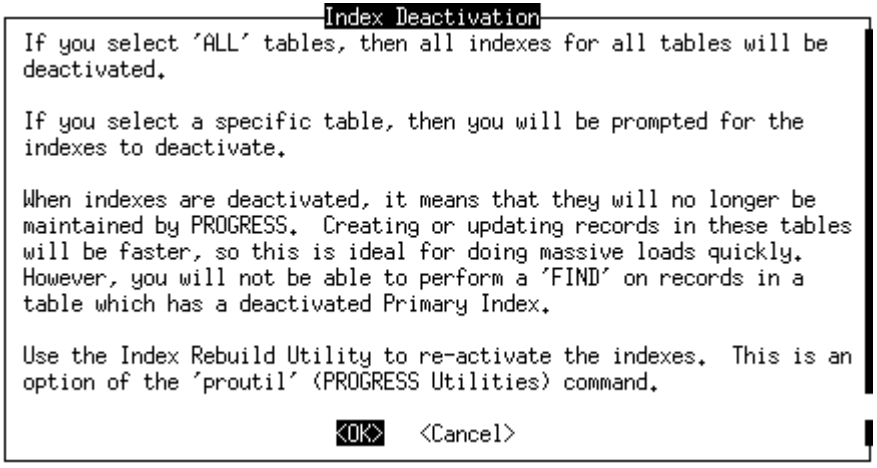


Figure 11–12: Index Deactivation Dialog Box

After you read the information, choose OK. Progress alphabetically lists all the tables defined for the database. Select the table for which you want to deactivate indexes or enter ALL to deactivate all your indexes. If you specified an individual table, Progress lists the indexes defined for that table. Select the indexes you want to deactivate, then press **GO**. Progress prompts you to verify your entry.

11.2.7 Utilities→ Information

Choose Utilities→ Information to display general information about the database and the Progress product you are using. Figure 11–13 shows the first portion of the report that appears.

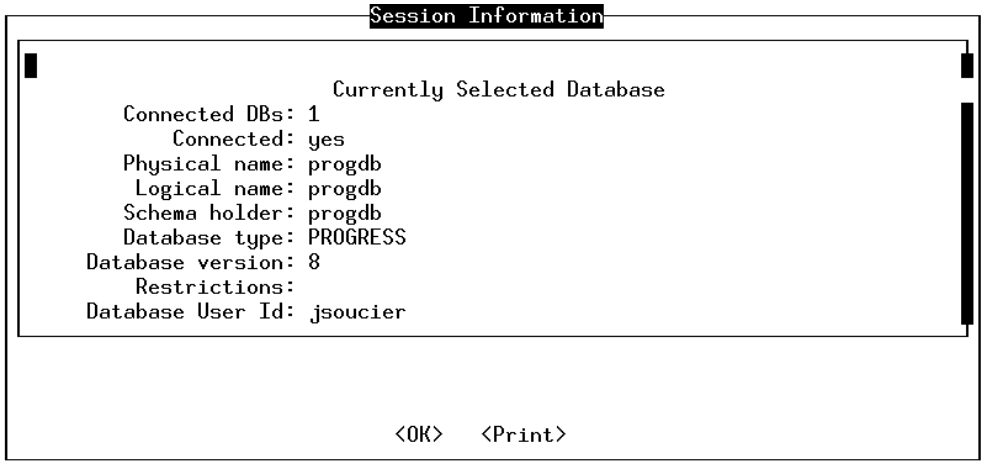


Figure 11–13: Session Information Dialog Box

To send the report to a file or printer, choose Print. Progress prompts you for the printer or filename, whether you want to append the report to an existing file, and the page length.

To exit the Information window, choose OK. Progress displays the video attribute settings that determine how the information appears on the monitor.

11.3 PRO/SQL Menu

Choose the PRO/SQL menu options to manage views created with Progress/SQL statements.

Table 11–6 describes the menu that appears when you choose this option. More detailed descriptions of the options follow the table.

Table 11–6: PRO/SQL Menu

PRO/SQL	
PRO/SQL View Report...	Lists information stored for views.
Dump as CREATE VIEW Statement...	Generates SQL CREATE VIEW statements.
Dump as CREATE TABLE Statement...	Generates SQL CREATE TABLE statements.

11.3.1 PRO/SQL→ PRO/SQL View Report

Choose this option to see the information about all SQL views defined for the working database.

NOTE: Views of non-Progress databases are treated as tables. To display information for these views, use the table reports option. See Chapter 8, "Database Menu Reference," for more information on generating a table report.

When you choose PRO/SQL→ PRO/SQL View Report, Progress alphabetically lists all the views defined for your Progress database. Select the views for which you want information. Figure 11–14 shows the first portion of an example view report.

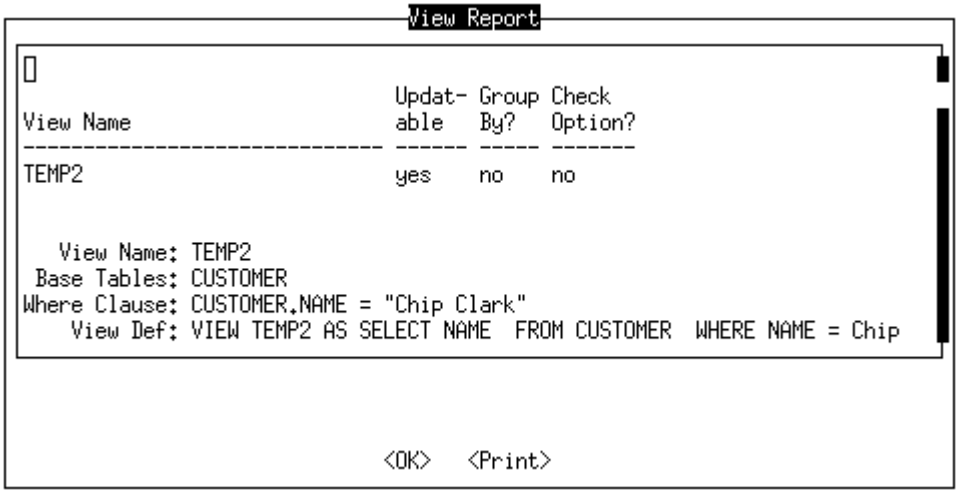


Figure 11–14: View Report

The View Report contains the following fields:

View Name

The name of the view.

Updatable

Whether the view is updatable.

Group By

Whether the view is defined with a GROUP BY clause.

Check Option

Whether the view is defined with a CHECK OPTION WHERE clause.

View Name

View name of the currently displayed view.

Base Tables

Tables from which the view gets its information.

Where Clause

The WHERE clause.

View Def

The SQL statement executed when defining the view.

Can-Read

Users who can read the rows of the view.

Can-Write

Users who can update records of the view.

Can-Create

Users who can create new records.

Can-Delete

Users who can delete records from the view.

Order

Order of the columns within the view.

View Column Name

Names of the columns that comprise the view.

Base Columns

Base table field names for each view column.

To send the report to a file or printer, choose Print. Progress prompts you for the printer or filename, whether you want to append the report to an existing file, and the page length.

11.3.2 PRO/SQL→ Dump as CREATE VIEW Statement

Choose PRO/SQL→ Dump as CREATE VIEW Statement to generate an SQL DDL program that contains CREATE VIEW statements equivalent to those originally used to define the selected view.

NOTE: Views of non-Progress databases are treated as tables. To display information for these views, use the table reports option. See the "Database→ Reports" section earlier in this manual for more information on generating a table report.

When you choose this option, Progress alphabetically lists all the views defined for your Progress database. Select the views for which you want make CREATE VIEW statements. A dialog box similar to the one shown in Figure 11–15 appears.

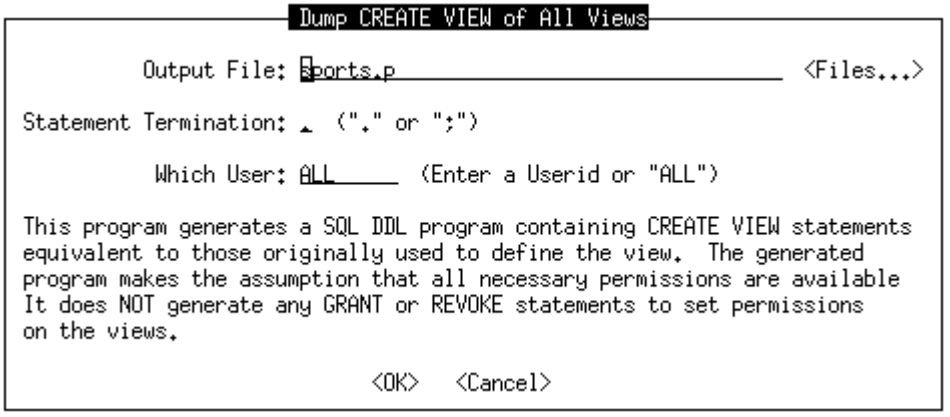


Figure 11–15: Dump CREATE VIEW Dialog Box

This dialog box contains the following user-interface elements:

Output File

Specifies the filename to which you want to write the CREATE VIEW statements. The default filename is either the view or database name with a .p extension.

Statement Termination

Specifies the statement termination character. The options are a period (.) or a semi-colon (;). The default value is a period (.).

Which User

If all views are selected, specifies all users or the User ID of the single user for whom you want to dump the views. The default value is All Users.

After generating the file, Progress prompts you to continue.

11.3.3 PRO/SQL→ Dump as CREATE TABLE Statements

Choose PRO/SQL→ Dump as CREATE TABLE and CREATE INDEX statements to generate an SQL DDL program that contains CREATE TABLE statements equivalent to those originally used to define the selected table.

NOTE: This option also works with non-Progress databases.

When you choose this option, Progress alphabetically lists all the tables defined for your database. Select the table for which you want to make CREATE TABLE statements. A dialog box similar to the one shown in Figure 11–16 appears.

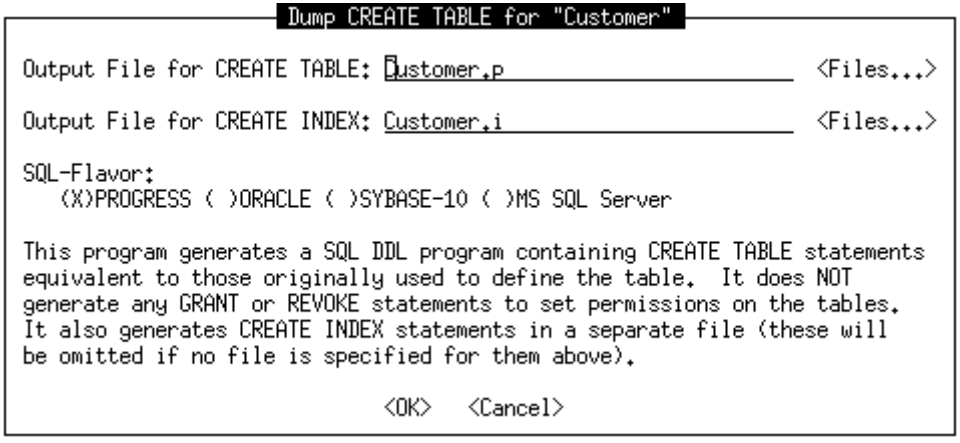


Figure 11–16: Dump CREATE TABLE Dialog Box

This dialog box contains the following user-interface elements:

Output File for CREATE TABLE

Specifies the filename to which you want to write the CREATE TABLE statements. The default filename for a single table is the *table-name*.p or *database-name*.p if you specified all tables.

Output File for CREATE INDEX

Specifies the filename to which you want to write the CREATE INDEX statements. The default filename for a single table is the *table-name*.i or *database-name*.i if you specified all tables. If you do not specify a filename, Progress writes the CREATE INDEX statements to the CREATE TABLE output file.

SQL-Flavor

Specifies the type of SQL you want to use.

After generating the file, Progress prompts you to continue.

11.4 Tools Menu

The Tools menu lets you access other Progress tools. See *Progress Basic Development Tools* for more information about the Tools menu.

Index

A

Abbr attribute
  Index Editor 9–34

Actv attribute
  Index Editor 9–33

Add field option
  Field Editor 9–13

Add new table option
  Schema menu 9–11

Add option
  Index Editor 9–35

Adding
  fields 9–14

Administrators
  security 10–15

Asc attribute
  Index Editor 9–34

ASCII option
  Export menu 10–16, 10–18

ASSIGN statement
  generating include files 2–14, 11–7

ASSIGN trigger 9–27

Attributes
  abbr 9–34
  actv 9–33
  asc 9–34
  BY-clause 10–18
  case-sensitive 9–25
  check CRC 9–27
  column-label 9–21
  CREATE TABLE statement 11–20
  CREATE VIEW statement 11–18
  cycle at limit 9–37
  databases 11–12
    connecting 8–7
    creating 8–5
  DataServer name 9–4, 9–37
  data-type 9–23
  decimals 9–24
  desc 9–26
  description 9–4
  disable triggers 10–18
  dump file 9–4
  export 10–17
  extent 9–24
  field 9–14 to 9–26
  field name 9–14, 9–33
  FORM statement 11–9
  format 9–14
  frozen 9–4
  help 9–26
  hidden 9–4
  import file 10–20
  increment by 9–37

index 9–32 to 9–34
initial 9–22
initial value 9–37
label 9–3, 9–8, 9–21
labeling 11–9
lower limit 9–37
mandatory 9–25
name 9–36
order 9–24
output file 11–9, 11–20
overrideable 9–27
owner 9–37
record size 9–4
seq 9–33
sequences 9–36 to 9–38
String 9–8
table type 9–3
tables 9–3
type 9–34
unique 9–32
upper limit 9–37
valexp 9–25
valmsg 9–26
WHERE-clause 10–18
word 9–33

Auto-connect lists
auto-connect records only option 10–5
editing 11–11

B

Binary dump and load 2–11

Block size 8–6

Bulk loader description files
creating 2–13

By column ranges option
Quoter Functions menu 11–6

By delimiter option
Quoter Functions menu 11–5

BY-clause
data export attribute 10–18

C

Can-Create
table permission 10–13

Can-Delete
table permission 10–13

Can-Dump
table permission 10–14

Can-Load
table permission 10–14

Can-Read
table permission 10–13

Can-Write
table permission 10–13

Case-sensitive attribute
Field Editor 9–25

Case-sensitivity
names 9–3

Changing passwords
Security menu 10–12

Character format
Field Editor 9–15

Character mapping 2–7, 10–4

Check CRC attribute
triggers 9–27

Collation tables option
dump data and definitions menu 10–5

Column ranges 11–6

Column-Label attribute
Field Editor 9–21

Components
index 9–23
view 9–22

Copying
fields 3–4

CRC code 8–16

CREATE
database events 9–7

CREATE TABLE statement 11–19
attributes 11–20
generating 7–2

CREATE VIEW statement 11–18
attributes 11–18
generating 7–2

Creating
auto-connect lists 11–11
bulk loader description files 2–13
databases 8–5
fields 3–2
include files 2–14
incremental data definition files 1–6, 10–6
indexes 4–2, 9–35
parameter files 1–7
reports 2–5
detailed table 2–5
field 3–6
index 4–6
quick table 2–6
sequence 5–5
SQL view 7–3
table relations 2–6
triggers 6–6
schema triggers 6–2
sequences 5–2
SQL programs 7–2
CREATE TABLE statements 7–2
CREATE VIEW statements 7–2
tables 2–2, 9–11
triggers
ASSIGN 6–3

Current database
selecting 8–4

Cycle at limit attribute
Sequence Editor 9–37

Cyclical redundancy check. *See* CRC code

D

Data
dumping 2–7
loading 2–9

Data Administration tool
loading database contents 2–9

Data definition files
creating an incremental file 1–6
dumping 1–6
loading 1–8

Data Dictionary
character
defining fields 3–2
examining field definitions 3–2
Database menu 8–3
DataServer menu 11–2
dumping
database contents 2–7
main window 1–2
menu bar 8–2
PRO/SQL menu 11–15
starting 1–2
Utilities menu 11–2

Data files
formatting 11–4

Database contents
dumping 2–7
loading 2–9

Database data
dumping 10–2
exporting 10–16
formats 10–16
ASCII option 10–18
dBASE definitions option 10–22
dBASE file contents option 10–19
delimited ASCII option 10–19
DIF option 10–19
fixed-length option 10–21
SYLK option 10–19

importing 10–19
loading 10–7
security 10–12
table contents (.d file) option 10–3

Database definitions
data definitions (.df file) option 10–3
dumping 10–2
loading 10–7

Database events
CREATE 9–7
DELETE 9–7
FIND 9–7
REPLICATION CREATE 9–7
REPLICATION DELETE 9–7
REPLICATION WRITE 9–7
WRITE 9–7

Database menu
Data Dictionary 8–3

Database types
database attribute 8–7

Databases
attributes 11–12
connecting 8–7
comparing 10–6
copying 8–6
creating 8–5
disconnecting 8–9
dumping
contents 2–7
empty 8–6
generating reports 8–9
information option 11–14
loading contents 2–9
partitioning 8–7
replacing 8–6
selecting 8–4
viewing session information 1–5

DataServer menu
Data Dictionary 11–2

DataServer name
Sequence Editor 9–37

DataServer name attribute 9–4

Data-Type attribute
field editor 9–23

Date format
Field Editor 9–19

dBASE definitions option
Import menu 10–22

dBASE file contents option
Import menu 10–19

Deactivating indexes 4–5, 11–14

Decimal format
Field Editor 9–16

Decimals attribute
Field Editor 9–24

DEFINE WORKFILE statement
generating include file 2–15

DEFINE WORKTABLE statement 11–10

DELETE
database events 9–7

Deleting
fields 3–5
indexes 4–5
sequences 5–3
tables 2–4, 9–11
triggers 6–5

Delimited ASCII option
Import menu 10–19

Delimiting
fields 11–5

Desc attribute
field editor 9–26

Description attribute 9–4

Detailed table reports 8–10

Dialog boxes
create database 8–5
database connect 8–7
detailed table report 8–11
dump CREATE TABLE 11–19
dump CREATE VIEW 11–18
dump data contents for some tables 10–4
edit data security 10–13
edit parameter file 11–3
export DIF 10–17
field triggers 9–27
find database file 8–6
fixed length field columns 10–22
freeze/unfreeze table 11–13
generate FORM 11–8
global field name change 9–31
import definitions 10–22
import DIF 10–20
import fixed length 10–21
index deactivation 11–14
output record formatting 10–19
parameter file comments 11–4
quick field report 8–13
quick index report 8–14
quick table report 8–12
quote by column 11–6
quote by delimiter 11–5
quote entire lines 11–5
reconstruct bad load records 10–9
report options 8–10
resequence order of fields 9–30
security administrators 10–15
select working database 8–4
sequence report 8–15
session information 11–15
SQL view report 8–15
table relations report 8–18, 8–19
table string attributes 9–8
table triggers 9–6
table validation 9–5
trigger code 9–8
trigger report 8–16
user report 8–17, 10–16
view-as phrase 9–28

DIF option
Export menu 10–16
Import menu 10–19

Disable triggers
data export attribute 10–18

Disconnecting
databases 8–9

Dump as CREATE TABLE statement option
PRO/SQL menu 11–19

Dump file attribute 9–4

Dump views of userid
CREATE VIEW statement attribute 11–19

Dumping
data and definitions 10–2
data definition files 1–6
database contents 2–7
SQL view file contents 7–3
SQL views 11–18
user table contents 2–8, 10–5, 10–9

E

Editing
data security 10–12
fields 9–11
indexes 9–31
parameter files 1–7
Progress auto-connect lists 11–11
sequences 9–36
user lists 10–10

Editor for parameter files option
Utilities menu 11–3

Entire lines option
Quoter Functions menu 11–4

Error files 10–9

Exit menu
Data Dictionary 8–19

Export
attributes 10–17 to 10–19

Export data option
Admin menu 10–16

Exporting
fields 10–18

Extent attribute
Field Editor 9–24

F

Field delimiter
data export attribute 10–19

Field Editor
go index option 9–29
Schema menu 9–11
triggers option 9–27
view as option 9–28

Field name attribute
Field Editor 9–14
Index Editor 9–33

Field reports 3–6

Field separator 10–19, 10–21

Fields
adding 9–13
adding to an existing table 9–14
changing field order 3–6
changing names globally 9–31
columns 11–6
copying 3–4
creating 3–2
defining 3–2
deleting 3–5
delimiting 11–5
editing 9–11
exporting 10–18
generating reports 8–12
modifying 3–3
reordering 9–30
triggers 9–27

Files
bulk loader description 2–13
formatting 11–4
include file
generating 2–14
generating DEFINE WORKFILE statements 2–15
parameters
creating 1–7
editing 1–7
SQL view
dumping contents 7–3
generating reports 7–3
loading contents 7–3

FIND
database events 9–7

Fixed-length option
Import menu 10–21

FORM statement
attributes 11–9
generating include files 2–14, 11–8

Format attribute
Field Editor 9–14

Formats
character 9–15
date 9–19
decimal 9–16
integer 9–16
logical 9–20

Formatting
data files 11–4
FORM statement attribute 11–9
text 10–18

Frame type
FORM statement attribute 11–9

Freezing and unfreezing tables 2–4, 11–13

Frozen attribute 9–4

Fully expand arrays
FORM statement attribute 11–9

Fully qualify names
  FORM statement attribute 11–9

G

Gateways. *See* DataServer

Generate include files option
  Utilities menu 11–7

Generating
  include files 2–14, 11–7
    ASSIGN statements 2–14
    DEFINE WORKFILE statement 2–15
    FORM statements 2–14
  reports 2–5
    database 8–9
    detailed table 2–5, 8–10
    field 3–6
    index 4–6
    quick field 8–12
    quick index 8–13
    quick table 2–6, 8–11
    sequence 5–5, 8–15
    SQL views 7–3, 8–14
    table relations 2–6, 8–18
    triggers 6–6, 8–16
    user 8–17
  SQL programs 7–2
    CREATE TABLE statements 7–2
    CREATE VIEW statements 7–2

Global field name change option
  Schema menu 9–31

GoIndex option
  Field Editor 9–29

H

Help attribute
  Field Editor 9–26

Hidden attribute 9–4

I

Import file attribute 10–20

Import menu 10–19 to 10–22
  fixed-length option 10–21

Include files
  generating 2–14, 11–7 to 11–11

Increment by attribute
  Sequence Editor 9–37

Incremental .df file
  creating 10–6

Index Editor
  accessing 9–29, 9–31
  add option 9–35

Indexes
  adding 9–35
  attributes 9–32 to 9–34
  creating 4–2
  deactivating 4–5, 11–14
  defining
    using multiple files 9–31
  deleting 4–5
  editing 9–31
  generating reports 4–6, 8–13
  modifying 4–4
  renaming 4–4

Information option
  Utilities menu 11–14

Initial attribute
  Field Editor 9–22

Initial value attribute
  Sequence Editor 9–37

Integer format
  Field Editor 9–16

J

Joins
table reports 8–18

L

Label attribute 9–3
Field Editor 9–21
FORM statement 11–10
table string 9–8

Labeling attribute
FORM statement 11–9

Loading
data and definitions 10–7
data definition files 1–8
database contents 2–9
reconstructed bad records 10–9
reconstructing bad records 2–10
sequences current values 10–9
SQL view file contents 7–3
user table contents 2–10

Locking
tables 11–13

Logical format
Field Editor 9–20

Logical names
database attribute 8–7

Lower limit attribute
Sequence Editor 9–37

M

Mandatory attribute
Field Editor 9–25

MAP. *See* Character mapping

Menu bar
Data Dictionary 8–2

Menus
Data Dictionary
Database 8–3
DataServer 11–2
PRO/SQL 11–15
Utilities 11–2

Microsoft Word option
Export menu 10–16

Modifying
auto-connect lists 11–11
fields 3–3
indexes 4–4
deactivating 4–5
sequences 5–3
table schema 9–3
tables 2–3
fields 3–3, 3–5, 3–6
primary index 4–4
renaming indexes 4–4
triggers 6–4

Multiple users
database attribute 8–8

N

Name
sequence attribute 9–36

Names
field 9–31

Naming
indexes 4–4

Non-Progress databases 11–11

O

Order attribute
Field Editor 9–24

Other UNIX-style parameters
database attribute 8–9, 11–12

Output file attribute 11–18, 11–20

Overrideable attribute
triggers 9–27

Overriding defaults
display order 9–24

Owner
Sequence Editor 9–37

P

Parameter files
creating 1–7
database attribute 8–8
editing 1–7, 11–3

Passwords
changing 10–12
database attribute 8–8

Performing
binary dump 2–11
binary load 2–13

Permissions
table 10–13

Physical database names
attribute 8–5

Primary indexes
changing 4–4

PRO/SQL menu
Data Dictionary 11–15

Q

Quick field reports 8–12

Quick index reports 8–13

Quick table reports 8–11

R

Reconstructing
bad load records 2–10, 10–9

Record end string
data export attribute 10–19

Record size attribute 9–4

Record starter string
data export attribute 10–19

Records
reconstructing 10–9

Renaming
indexes 4–4

Reordering fields 9–30

Replication
CREATE 9–7
DELETE 9–7
WRITE 9–7

Reports
detailed table 2–5, 8–10
field 3–6
generating 2–5
index 4–6
PRO/SQL view 11–16
quick field 8–12
quick index 8–13
quick table 2–6, 8–11
sequence 5–5, 8–15
SQL view 7–3, 8–14
table relations 2–6, 8–18
trigger 6–6, 8–16
user 8–17, 10–16

Reports menu 8–9

S

Security
  administrators 10–15
  changing passwords 10–12
  editing data 10–12
  editing user lists 10–10
  permissions 10–13
  user reports 10–16

Security option
  Admin menu 10–10

Selecting
  working database 8–4

Seq attribute
  Index Editor 9–33

Sequel statements. *See* SQL

Sequence Editor 9–36
  attributes 9–36 to 9–37
  options 9–38

Sequences
  creating 5–2
  deleting 5–3
  dumping
    definitions 10–5
    values 10–6
  dumping and loading 5–5
  generating reports 5–5, 8–15
  loading current values 10–9
  modifying 5–3

_seqvals.d file 5–5

Session information 11–14

SQL
  dumping as CREATE VIEW statements
    option 11–18
  generating reports 8–14
  view report option 11–16

SQL views
  dumping 7–3, 10–5
  generating reports 7–3
  loading 7–3, 10–8

Starting
  Data Dictionary 1–2

Statement termination character 11–19

Statements
  ASSIGN 2–14, 11–7
  CREATE TABLE 7–2, 11–19
  CREATE VIEW 7–2, 11–18
  DEFINE WORK-TABLE 2–15, 11–10
  FORM 2–14, 11–8, 11–9

String attributes 9–8

SYLK option
  Export menu 10–16
  Import menu 10–19

T

Table reports
  detailed 2–5
  quick 2–6
  relations 2–6, 8–18

Table type attribute 9–3

Tables
  access restrictions 10–14
  adding 9–11
  attributes 9–3
  changing field order 3–6
  collation 10–5
  converting 10–5
  creating 2–2
  deleting 2–4, 9–11
  dumping as CREATE TABLE statements
    11–19
  freezing and unfreezing 2–4, 11–13
  generating reports 8–10
  hidden 9–4

loading
reconstructed bad records 2–10
modifying 2–3, 9–3
adding fields 3–2
copying fields 3–4
deleting fields 3–5
fields 3–3
naming 9–3
owners 9–4
permissions 10–13
string attributes 9–8
translating 10–5
triggers 9–6
user tables
dumping contents 2–8
loading contents 2–10
validating 2–2, 9–5

Text formats
exporting 10–18

Translation tables 10–5

Trigger locations
database attribute 8–8

Triggers 9–6
attributes 9–27
creating
ASSIGN 6–3
schema 6–2
database events
CREATE 9–7
DELETE 9–7
FIND 9–7
REPLICATION CREATE 9–7
REPLICATION DELETE 9–7
REPLICATION WRITE 9–7
WRITE 9–7
deleting 6–5
disabling during export 10–18, 10–21
field 9–27
generating reports 6–6, 8–16
modifying 6–4

Type attribute
Index Editor 9–34

U

Unique attribute
Index Editor 9–32

Upper limit attribute
Sequence Editor 9–37

User access 10–16

User IDs
database attribute 8–8

User lists
editing 10–10

User report option
Security menu 10–16

User table contents
dumping 10–5, 10–9

Users
generating reports 8–17

Utilities menu
Data Dictionary 11–2

V

Valexp attribute
Field Editor 9–25

Validation
FORM statement attribute 11–9

Validation criteria 2–2, 9–5

Validation messages
table string attributes 9–9

Valmsg attribute
Field Editor 9–26

Variable block size 8–6

View as option
Field Editor 9–28

W

WHERE-clause
  data export attribute 10–18

Windows
  add new table 9–11
  create incremental .df file 10–6
  edit user lists 10–11
  editing Progress auto-connect lists 11–12
  field add option 9–14
  field editor 9–12
  goindex option 9–30
  index add option 9–35
  Index Editor 9–32
  modify table 9–3
  Sequence Editor 9–36

Word attribute
  Index Editor 9–33

WordPerfect option
  Export menu 10–16

WordStar option
  Export menu 10–16

Working databases
  defined 1–2
  selecting 8–4

WRITE
  database events 9–7