PROGRESS®

FAST TRACK USER'S GUIDE

Printed in U.S.A.

June, 1990

CONTENTS

Figures

Tables

Preface

This reference manual introduces you to PROGRESS FAST TRACK. FAST TRACK is a versatile, menu driven application generator for the PROGRESS environment.

In this book, the term PROGRESS refers to the PROGRESS 4GL/RDBMS (formerly Full PROGRESS). The PROGRESS Application Development System consists of the PROGRESS 4GL/RDBMS and the PROGRESS FAST TRACK Application Builder.

AUDIENCE

This book serves as a developer's guide and reference manual for PROGRESS FAST TRACK. It is suggested that even experienced PROGRESS developers read the *PROGRESS FAST TRACK Tutorial* before reading this book. Combined, the two books provide the information you need to use FAST TRACK effectively.

FAST TRACK is designed for several types of users. PROGRESS developers can use FAST TRACK to quickly design and develop applications. They can also use it as a tool to prototype applications. Users won't always use FAST TRACK to do full-scale application development, however. Users with little or no programming experience can use FAST TRACK to expand existing applications. FAST TRACK supplies a friendly interface to perform a variety of database operations and display results immediately. This makes FAST TRACK a useful tool for the end-user.

This reference manual addresses the needs of all these audiences. It provides an introduction to the concepts and capabilities of FAST TRACK and explains the FAST TRACK interface and features. If you are using FAST TRACK to do full-scale application development, acquaint yourself with the basic PROGRESS concepts introduced in the *PROGRESS Language Tutorial* before reading this reference manual.

ORGANIZATION OF THIS BOOK

The organization of this book is important to understand. Begin by reading Chapter 1 to get an overview of the PROGRESS FAST TRACK interface and components. This book is organized as follows:

Chapter 1 — PROGRESS FAST TRACK: An Overview

Provides an overview of FAST TRACK, and summarizes the options on the FAST TRACK Main Menu, details the use of the `prodb` and `proft` commands to start up FAST TRACK, and describes the various modules in FAST TRACK.

Chapter 2 — FAST TRACK Interface

Describes the FAST TRACK menu and window interface, default settings, and other system information, including function keys and how to get help from FAST TRACK. This chapter serves as a reference for the entire book.

Chapter 3 — The Menu Editor

Describes how to create menus and generate menu code for use in PROGRESS applications.

Chapter 4 — The Screen Painter

Describes how to create, design, and modify forms and generate code for use in PROGRESS applications.

Chapter 5 — The Report Writer

Explains the different report types you can create while using the Report Writer editor. Shows how to create simple and complex reports from information in a database, and generate code to create reports in PROGRESS applications.

Chapter 6 — The QBF Generator

Explains query-by-form (QBF) operations that selectively display, add, edit, or delete records in database files, and how to generate code that can be used to perform these operations in PROGRESS applications.

Chapter 7 — Maintenance

Describes how to delete a menu, report, QBF or form from the database, dump FAST TRACK data files into files in your directory, load FAST TRACK data files back into the FAST TRACK database, deploy your application to a deployment directory, compile your application, define runtime access privileges, and more.

Chapter 8 — Database Programming With FAST TRACK

Provides an introduction to "multi-database" programming with PROGRESS FAST TRACK and the PROGRESS 4th generation language (4GL).

TYPOGRAPHICAL CONVENTIONS

This document uses the following typographical conventions:

- **Bold typeface** indicates filenames and commands you type.. It also emphasizes important points.
- *Italic typeface* indicates a parameter or argument you supply. It also introduces new terms and manual titles.
- `Typewriter typeface` indicates system output and PROGRESS procedures. It also highlights file names, field names, command names, and menu options in text.

The following typographical conventions are used to represent keystrokes.

- A box labeled with the name of the key represents a single keystroke:

  [CTRL] [↑] [O] [HELP] [GO] [END]

- When you must hold down one key while pressing another to issue a command, then the two keys are connected by a hyphen (-). For example:

  [CTRL]-[O]

- When you must press two keys separately in sequence, they appear in the order in which you enter them.

  [ESC] [C]

- In some cases, such as [GO], [OPTIONS], [HELP] or [CHOICES], the name of the key does not match the label on the key cap. In these cases, the label used on most terminals is placed in parentheses after the key name. For example:

  [GO] (F1) [OPTIONS] ([CTRL]-[O]) [CHOICES] ([ESC] [C])

  Key names are used instead of labels because labels may differ from terminal to terminal.

- Some menu options appear in capital typewriter letters. When one option displays another menu from which a choice must be made, the menu options are separated by a right arrow (→) to represent the movement from one menu to another. For example:

  `OTHER→MAIN-MENU  INSERT→FIELD`

NOTE: If you are running FASTTRACK on a DOS system, press the [ALT] key wherever this book refers to the [ESC] key. Also, when using the [ALT] key, you must hold it down while pressing an accompanying key to use a command.

OTHER USEFUL PUBLICATIONS

The following is a list of other publications from Progress Software Corporation which you may find useful:

PROGRESS FAST TRACK Tutorial

Provides a " how-to" guide for using PROGRESS FASTTRACK. This manual is designed to lead both a novice end-user and an application developer through the major FAST TRACK features.

PROGRESS Installation Notes

Contains step-by-step instructions for installing PROGRESS. Describes the prerequisites and procedures to get PROGRESS up and running on your machine.

PROGRESS Test Drive

Introduces new users to PROGRESS through a sporting goods distributor's inventory and order processing application.

PROGRESS Language Tutorial

Provides a "how-to" guide to PROGRESS fundamentals, designed for both novice and experienced programmers.

PROGRESS Programming Handbook

Details advanced PROGRESS programming techniques. Provides more detailed information about application development with PROGRESS.

PROGRESS Language Reference

A detailed library of information on a number of PROGRESS topics. Provides descriptions and examples for each statement, function, phrase, and operator in the PROGRESS language.

System Administration I: Environments

Explains the DOS, UNIX, VMS, and BTOS/CTOS concepts required to run PROGRESS and provides information about running PROGRESS on networks.

System Administration II: General

Describes PROGRESS limits, disk and memory requirements, startup and shutdown procedures, backing up and restoring databases, and PROGRESS utilities. It also provides information about security administration, using multi-volume databases, and Roll Forward Recovery.

Pocket PROGRESS

Lets you quickly look up information about the PROGRESS language or programming environment.

Developer's Toolkit Manual

Explains how to use the PROGRESS Developer's Toolkit, a set of tools used to prepare PROGRESS applications for distribution.

Database Gateways Guide

Provides information about the how to use the PROGRESS 4th generation programming language on different relational database management systems other than PROGRESS RDBMS.

3GL Interface Guide

Supplies information about the PROGRESS Host Language Call (HLC), embedded SQL, and the Host Language Reference (HLI). This manual also contains information on how to use the PROBUILD utility.

Chapter 1
PROGRESS FAST TRACK: An Overview

FAST TRACK is a menu-driven application that acts as an interface to the PROGRESS Fourth Generation Language (4GL). The FAST TRACK software consists of a set of editing and maintenance modules that let you quickly develop and deploy end-user applications.

In industry terms, FAST TRACK falls into the category of a computer-aided software engineering (CASE) product. The chief aim of a CASE tool is to develop applications level software. The fundamental benefit of FAST TRACK is that it generates code that you ordinarily would write yourself.

PROGRESS users and developers at all levels can benefit by using FAST TRACK, which itself is written in PROGRESS. Here are some likely scenarios for developers and end-users:

- An experienced PROGRESS programmer, who doesn't want to spend the time coding menus, uses FAST TRACK's Menu Editor to automatically produce menu code.
- An applications developer, who is a new PROGRESS user, can develop the bulk of an application in FAST TRACK, and begin to learn PROGRESS at the same time.
- An end-user, who puts a premium on easily formatted output, can use FAST TRACK's Report Writer to develop custom reports.

This chapter introduces the following topics:

- FAST TRACK components and utilities.
- Developing an application.
- FAST TRACK environments.
- Creating a database.
- Starting FAST TRACK.

- Using fast track with a PROGRESS database.
- Moving between FAST TRACK and PROGRESS.

1.1 FAST TRACK COMPONENTS AND UTILITIES

FAST TRACK is a multi-level software program. When you load and run FAST TRACK, the Main Menu appears on your terminal screen. From the Main Menu, you select options that allow you to create PROGRESS procedures, generate reports, and perform database maintenance.

Each Main Menu option itself provides menus and submenus from which you select options to perform a given task. Additionally, several of the menu options let you automatically enter a different editor or utility, without backing out to the Main Menu.

You can use various key sequences to select options from the Main Menu. Chapter 2 describes option selection methods in detail. For now, refer to Figure 1-1 as you read about the options available on the Main Menu.

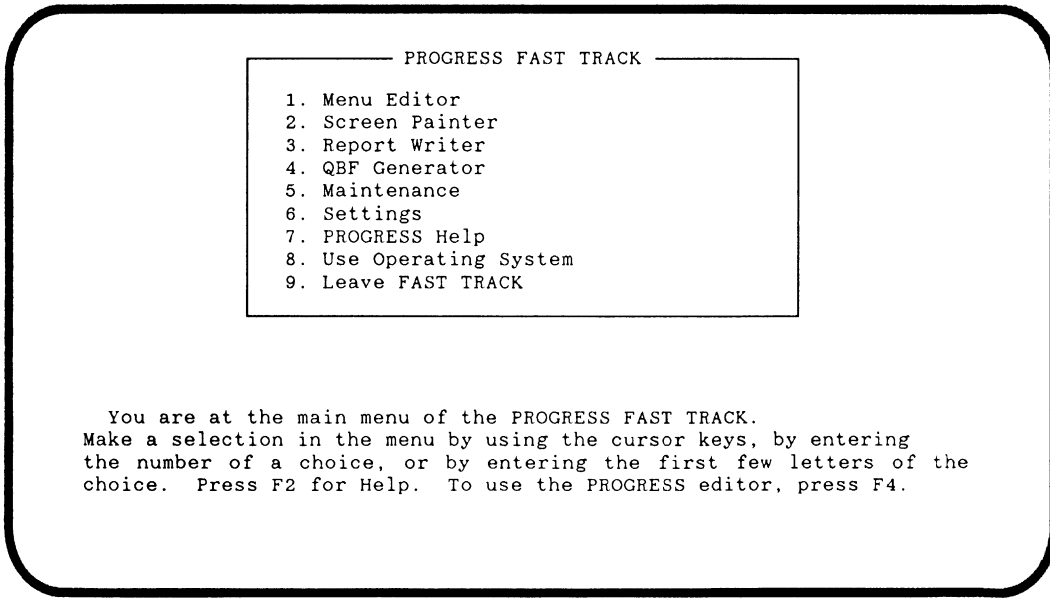


Figure 1-1: FAST TRACK Main Menu

Options 1 through 3 on the Main Menu invoke the following FAST TRACK editors:

- **Menu Editor.** Creates menus and submenus, and links executable actions to menu choices. You can associate a menu choice with a PROGRESS procedure, a QBF, a report, or another menu.

- **Screen Painter.** Creates screen forms that serve as the data entry area for the end-user. Forms consist of fields, variables and text. The Screen Painter lets you place these objects on the screen as you want them to appear to the end-user. Note that in the software, the Screen Painter is sometimes referred to as the Forms Editor.

- **Report Writer.** Creates and designs reports using the fields in your database files. You can use one or more database files in a report and choose the fields that appear on the report. The Report Writer supplies a number of editing functions that allow you to design your report any way you like.

Options 4 through 9 on the Main Menu invoke the following FAST TRACK modules and utilities:

- **QBF Generator.** Create query-by-form procedures to access database information through forms rather than through a query language such as PROGRESS. If a form does not exist, the QBF Generator can create a default form for the QBF.

- **Maintenance.** Delete a menu, report, QBF or form from the database; dump FAST TRACK data files into files in a specified directory; load FAST TRACK data files back into the FAST TRACK database; deploy an application; compile an application; define runtime access privileges for menus, menu options, procedures, QBFs, and reports; maintain output devices; and produce development reports.

- **Settings.** Display the control key and function key sequences used to perform commands in the Menu Editor, Screen Painter, Report Writer, and QBF Generator.

- **PROGRESS Help.** Access the PROGRESS Help system from FAST TRACK. Once in the PROGRESS Help system, you can get help information on PROGRESS statements, functions, operators, keywords, and more. You can also access the PROGRESS Data Dictionary, access the operating system, list the file names in the current directory, or run a PROGRESS program.

- **Use Operating System.** Access your operating system, without actually leaving FAST TRACK.

1.2 DEVELOPING AN APPLICATION

Even with little PROGRESS programming knowledge, you begin application program development with FAST TRACK. In fact, you can develop applications with menus, QBFs and reports from start to finish using the various FAST TRACK editors and QBF generator.

An *application* is a collection of data definitions, menus, forms, reports, and procedures that a user needs to perform a set of tasks. *Data definitions* determine the kinds of information that the application uses. *Menus* organize the tasks into a logical structure. *Forms* lay out the data according to different users' needs or the requirements of a specific task. *Reports* organize and summarize the data. *Procedures* define the steps taken to perform the individual tasks.

Figure 1-2 shows how an application is built from PROGRESS procedures you create with PROGRESS and the menus, forms, reports, and QBF procedures you create with FAST TRACK, combined with the data definitions.

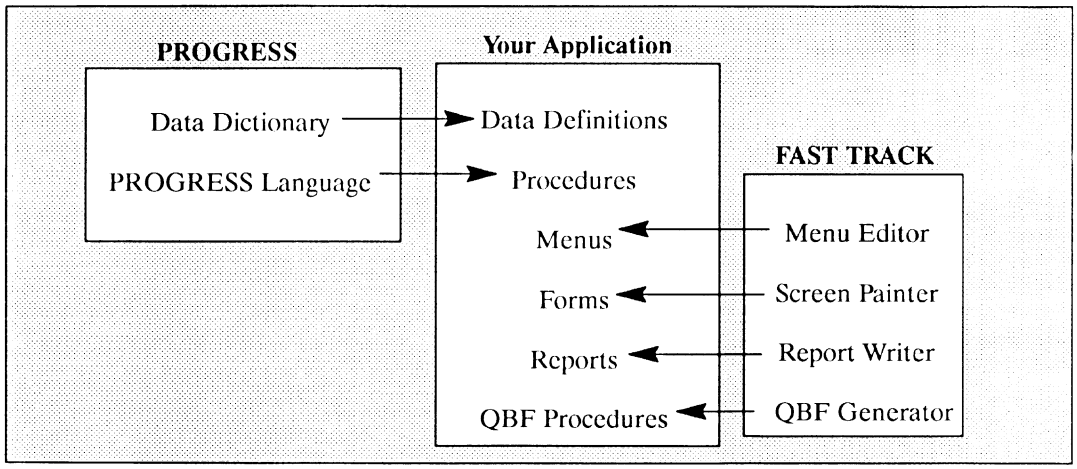


Figure 1-2: How an Application is Built with PROGRESS and FAST TRACK

Developing an application using both PROGRESS and FAST TRACK can be broken down into the following series of steps:

1. Create a new application database from either a FAST TRACK **empty** database or a copy of an existing PROGRESS database. If you use a PROGRESS database, however, youmust convert it to a FAST TRACK database (see section 1.5).

   With the PROGRESS Data Dictionary, define the files, fields, and indexes for your application. (Before you modify the schema, you must leave FAST TRACK (press the END (F4) key from the menu) and start the dictionary from Progress HELP (press HELP (F2) and choose option d from the PROGRESS Help menu). After exiting the dictionary, re-enter FAST TRACK by typing **run ft.p** and pressing GO (F1).

2. Load FAST TRACK from the operating system prompt (see section 1.4).

3. With the Menu Editor, build an application's basic menu structure (see Chapter 3).

4. With the Screen Painter, design an application's forms (see Chapter 4).

5. With the Report Writer, design and format an application's reports and specify the information to be included (see Chapter 5).

6. With the QBF Generator, produce QBF procedures for performing file maintenance and data retrieval with the forms (see Chapter 6).

7. Using the tools on the Maintenance menu, compile and deploy your application (see Chapter 7).

1.3 FAST TRACK ENVIRONMENTS

Because FAST TRACK is written in PROGRESS, you need PROGRESS on your system to execute FAST TRACK as well as any applications created with FAST TRACK. This section details the functionality of the various versions of FAST TRACK and PROGRESS.

To develop an application in any version of FAST TRACK, you must use the PROGRESS Data Dictionary to define new files and fields for your FAST TRACK database. In fact, to create a FAST TRACK database, you use the `prodb` utility in the same manner as you do to create a PROGRESS database.

Before you begin developing a FAST TRACK application, ensure that your FAST TRACK and PROGRESS products have the functionality to produce the kind of application you want.

NOTE: If you intend to deliver a Fast Track application to run on hardware and/or operating systems that are different from your own, or if you intend to compile your application system at a site without PROGRESS 4GL/RDBMS installed, you need the PROGRESS Developer's ToolKit to encrypt and/or compile your code.

Table 1-1 lists the functionality for the various combinations of FAST TRACK and PROGRESS.

Table 1–1: FAST TRACK Functionality from the Developer Perspective

	FAST TRACK Product			
	FAST TRACK	**Query Report**	**Run-time Utilities**	**No FAST TRACK**
PROGRESS 4GI/RDBMS	Full development	Reports	No development	No development
PROGRESS Query Report	Full development: except for update, delete in QBF	Reports	No development	No development
PROGRESS Run-time	No development	No development	No development	No development

Applications generated by FAST TRACK can be run by PROGRESS Run-time, PROGRESS Query/Report, and PROGRESS 4GL/RDBMS. Note that the different PROGRESS products support different levels of FAST TRACK functionality, depending on the installed version of FAST TRACK. Table 1–2 lists the user functionality for the various combinations of FAST TRACK and PROGRESS.

Table 1–2: FAST TRACK Functionality from the User–Only Perspective

	FAST TRACK Product			
	FAST TRACK	**Query Report**	**Run–time Utilities**	**No FAST TRACK**
PROGRESS 4GL/RDBMS	Full functionality	Full functionality	Full functionality	Reports; no RT security output control
PROGRESS Query Run–time	Full functionality	Full functionality	Full functionality	Reports; no RT security output control
PROGRESS Run–time	Full functionality; except QBF qualification, new include files, page size in output control option	Full functionality; except QBF qualification, new include files, page size in output control option	Full functionality; except QBF qualification, new include files, page size in output control option	Reports; no RT security, output control

1.4 CREATING A DATABASE

The first step in running FAST TRACK is to create a database. Ensure that your current directory is not the same directory that contains the FAST TRACK software. Typically, FAST TRACK resides in one of the following directories:

On UNIX `/usr/dlcft`

On DOS `\dlcft`

On VMS `[dlcft]`

On BTOS `[sys]<dlc>`

Use the `prodb` command to create your own empty FAST TRACK database. In the following examples, the symbolic name *dbname* represents the filename that you choose for your database.

Table 1-3: Database Copy Command

Operating System	To Create A Copy Of The "Empty" FAST TRACK Database
UNIX	`prodb` *db-name* `/usr/dlcft/emptyft`
DOS	`prodb` *db-name* `\dlcft\emptyft`
VMS	`PROGRESS/CREATE` *db-name* `[dlcft]emptyft`

BTOS/CTOS	`PROGRESS Create Database`
	`New Database Name` *db-name*
	`Copy From Database Name emptyft`

1.5 STARTING FAST TRACK

To start FAST TRACK using your database on UNIX and DOS, use the `proft` command:

Table 1-4: FAST TRACK Single-user Startup Command

Operating System	FAST TRACK Startup Command
UNIX	`proft` *db-name*
DOS	`proft` *db-name*
VMS	`@PROFT` *db-name*
BTOS/CTOS	`FAST TRACK 4GL` `[Options] -1` *db-name*

To start multi-user FAST TRACK using your database on UNIX, you must first use the PROGRESS `proserve` command to start a multi-user server for the database:

Table 1–5: PROSERVE Command

Operating System	PROSERVE Command
UNIX	`proserve` *db–name* `[options]`
DOS	`proserve` *db–name* `[options]`
VMS	`PROGRESS/MULTI-USER=START_SERVER` *db–name*

BTOS/CTOS	`PROGRESS Server`
	`Database Name` *db–name*
	`[Options]`

For more information about the `proserve` command, consult Chapter 2 of the *System Administration II: General* book.

Once you have started a server for a database, then you can use the `mproft` command to start a FAST TRACK multi-user session using that database:

Table 1–6: FAST TRACK Multi-user Startup Command

Operating System	FAST TRACK Startup Command
UNIX	`mproft` *db–name*
DOS	`mproft` *db–name*
VMS	`@MPROFT` *db–name*
BTOS/CTOS	`FAST TRACK 4GL` `[Options]` *db–name*

1.6 USING FAST TRACK WITH A PROGRESS DATABASE

If you want to run FAST TRACK on an existing PROGRESS (version 4.2E or later) database, you must convert it to a FAST TRACK database using the `convft` utility supplied with the FAST TRACK software. You need to perform this conversion because FAST TRACK and PROGRESS use different database file structures. To use `convft`, follow these steps:

1. Ensure that `convft` is located in a directory in your operating system path. If you installed FAST TRACK using the installation program, `convft` is automatically in your system path.
2. Use the appropriate command to access the `convft` utility on your system.

Table 1–7: FAST TRACK convft Command

Operating System	FAST TRACK convft Command
UNIX	`convft` *db–name*
DOS	`convft` *db–name*
VMS	`@CONVFT` *db–name*
BTOS/CTOS	FAST TRACK Convert Database Database Name *db–name*

You then see the following prompt:

```
Script convft is used to convert a non-FAST TRACK database into a
FAST TRACK one. If you do not wish to continue, press ^C anytime.
If convft does not finish properly for any reason, your database
might fail to run under PROGRESS.
You MUST have your database backed up before running convft.
Have you done this (y/n)?
```

Figure 1–3: System Prompt for the convft Command.

3. If you have not previously backed up your database, you should type **N** and do a backup immediately. If you enter **Y**, the following messages appear:

```
mydb is not a FAST TRACK database.
Loading FAST TRACK schema into mydb.
Conversion beginning.  Please await success confirmation...
Successful V6 FAST TRACK conversion completed.
Loading FAST TRACK data into mydb
Convft script completed.
```

Figure 1–4: Conversion Message for the convft Command.

4. When the database is loaded, control returns to the operating system. You can then execute the `proft` command to start up FAST TRACK using the converted database, which retains its original name. If you are running multi-user FAST TRACK, execute the `proserve` and `mproft` command instead.

1.7 MOVING BETWEEN FAST TRACK AND PROGRESS

If you want, you can access the PROGRESS editor directly from FAST TRACK. Additionally, you can use the PROGRESS editor to run QBF procedures or view code generated in FAST TRACK.

To access the PROGRESS editor from FAST TRACK, press the END (F4) key at the FAST TRACK main menu. To return to the FAST TRACK Main Menu from the PROGRESS editor, enter the procedure shown in Figure 1–5.

```
RUN ft.p.
```

Figure 1–5: RUN Statement in Progress Editor

Press the GO (F1) key and the FAST TRACK Main Menu appears.

Chapter 2
The FAST TRACK Interface

This chapter describes the FAST TRACK default key settings and introduces you to the menu and window system used by FAST TRACK. Additionally, it introduces you to the command menu used in the Menu Editor, Screen Painter, and Report Writer modules. The following list summarizes the topics covered in this chapter:

- Default Settings — show you the default key sequences associated with FAST TRACK commands.
- Menus — a multi-level system of horizontal and vertical menus that provide access to the FAST TRACK development tools and other utilities.
- Windows — screen areas in which you supply information and program data as you develop a FAST TRACK application.
- Command Menu — the horizontal menu that appears at the bottom of your screen in the Menu Editor, Screen Painter, and Report Writer modules.

2.1 DEFAULT SETTINGS

The `Settings` option on the Main Menu shows you the key sequences you use to invoke menus, prompts and internal utilities. These key sequences include standard editing keys and function keys as well as special label keys such as the [CHOICES] key.

Tables 2-1 and 2-2 list the default function key and editing key assignments. In Table 2-1, the column referred to as Label identifies the operation associated with a given key. These labels appear throughout this manual as well as in the *PROGRESS FAST TRACK Tutorial*.

Table 2-1: Function Key Defaults

Label	Key	Action
GO	(F1)	Runs the current menu option or accepts an entered field value.
HELP	(F2)	Enters the FAST TRACK Help system, which provides information about using FAST TRACK. To exit from the Help system, press END (F4).
MODE	(F3)	Changes between insert and overstrike modes. In insert mode, the character you type is inserted at the current cursor position. The cursor and the character under it move one space to the right. In overstrike mode, the character you type *replaces* the character at the current cursor position. Overstrike is the default mode.
END	(F4)	Exits from the current menu, window, or operation. If you are in the middle of a database operation such as updating a record, the changes are not made. Pressing END (F4) from the Main Menu invokes the PROGRESS editor.
CLEAR	(F8)	Backs up one command menu level.
INSERT	(F9)	Inserts a new line above the current line.
DELETE	(F10)	Removes the current line.

Table 2-2 lists additional key assignments.

Table 2-2: Other Editing Keys

Key	Action
RETURN	Moves the cursor to the next field. If on the last field, then the cursor moves to the next line.
BACKSPACE	Deletes the character to the left of the cursor and moves the cursor to the position of the deleted character.
↓ and ↑	Move the cursor one line at a time in the direction of the arrow.
→ and ←	Move the cursor one character position at a time in the direction of the arrow.
DEL	Deletes the character that is under the cursor and moves the characters to the right of the cursor one position to the left.
TAB	Moves the cursor to the next field or label.
BACKTAB	Moves the cursor to the previous field or label.

2.2 MODIFYING KEY ASSIGNMENTS

Experienced PROGRESS programmers can modify key settings and sequences for UNIX, BTOS, and VMS systems. By redefining key assignments, FAST TRACK can support terminals that require different key settings. For example, if your terminal uses the F1 key for a hardware specific function, you could reassign the GO key to a different key.

To redefine key assignments on UNIX, BTOS, and VMS systems, you must edit the `protermcap` file. For detailed information on editing `protermcap`, refer to Appendix E of this manual and the *Programming Handbook*.

FAST TRACK is flexible in the way it obtains keyboard input. It gives you three ways to enter command related input:

- Pressing a function key.
- Pressing the CTRL key simultaneously with another key. For example, OPTIONS (CTRL-O) invokes the horizontal command menu, which is explained later in this chapter.

- Pressing and releasing the [ESC] key and then pressing a second key. On DOS systems, however, this sequence is reproduced by pressing the [ALT] key simultaneously with another key.

Figure 2-1 shows the default key assignments and sequences for UNIX and VMS systems.

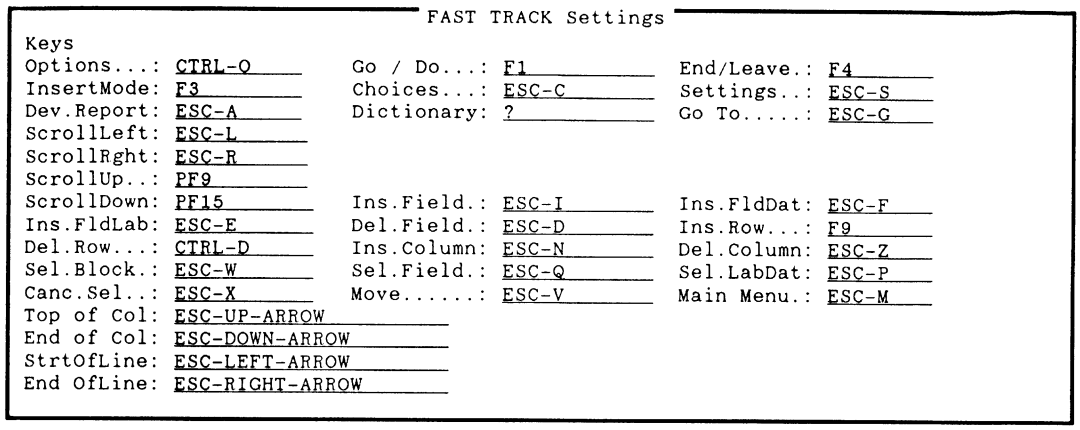


Figure 2-1: UNIX and VMS Key Settings

NOTE: On VMS, CTRL-O is replaced by ESC-O.

On a DOS system, FAST TRACK makes use of extended key sequences defined by the operating system. Figure 2-2 shows the default key assignments and sequences for DOS systems.

```
                    FAST TRACK Settings
Keys
Options...: ALT-O       Go / Do...: F1        End/Leave.: ESC
InsertMode: INS         Choices...: ALT-C     Settings..: ALT-S
Dev.Report: ALT-A       Dictionary: ?         Go To.....: ALT-G
ScrollLeft: ALT-L
ScrollRght: ALT-R
ScrollUp..: PAGE-UP
ScrollDown: PAGE-DOWN   Ins.Field.: ALT-I     Ins.FldDat: ALT-F
Ins.FldLab: ALT-E       Del.Field.: ALT-D     Ins.Row...: F9
Del.Row...: F10         Ins.Column: ALT-N     Del.Column: ALT-Z
Sel.Block.: ALT-W       Sel.Field.: ALT-Q     Sel.LabDat: ALT-P
Canc.Sel..: ALT-X       Move......: ALT-V     Main Menu.: ALT-M
Top of Col: ALT-T
End of Col: ALT-B
StrtOfLine: CTRL-LEFT
End OfLine: CTRL-RIGHT
```

Figure 2-2: DOS Key Settings

NOTE: The key sequences used in DOS systems vary slightly from those used in UNIX systems. This manual, however, uses the UNIX conventions. DOS users can translate most key sequences by pressing [ALT] instead of [ESC].

Figure 2-3 shows the default key assignments and sequences for BTOS systems.

```
                    FAST TRACK Settings
Keys
Options...: CODE-O          Go / Do...: F1          End/Leave.: CANCEL
InsertMode: OVERTYPE        Choices...: CODE-C      Settings..: CODE-S
Dev.Report: CODE-SHIFT-T    Dictionary: ?           Go To.....: CODE-SHIFT-G
ScrollLeft: SHIFT-LEFT-ARROW
ScrollRght: SHIFT-RIGHT-ARROW
ScrollUp..: PREV-PAGE
ScrollDown: NEXT-PAGE       Ins.Field.: CODE-I      Ins.FldDat: CODE-SHIFT-F
Ins.FldLab: CODE-SHIFT-L    Del.Field.: CODE-J      Ins.Row...: F9
Del.Row...: F10             Ins.Column: CODE-SHIFT-C Del.Column: CODE-SHIFT-D
Sel.Block.: CODE-SHIFT-A    Sel.Field.: CODE-SHIFT-B Sel.LabDat: CODE-K
Canc.Sel..: CODE-CANCEL     Move......: MOVE        Main Menu.: CODE-M
Top of Col: CODE-UP-ARROW
End of Col: CODE-DOWN-ARROW
StrtOfLine: CODE-LEFT-ARROW
End OfLine: CODE-RIGHT-ARROW
```

Figure 2-3: BTOS Key Settings

2.3 WINDOWS

All FAST TRACK modules use prompts and windows, into which you enter system information and program data. FAST TRACK sizes a window just large enough to display the required data. For example, if you only need to supply a filename, FAST TRACK displays a relatively small window.

When you select a module from the Main Menu, FAST TRACK immediately displays the module's initialization window. A prompt telling you exactly what type of information is required also appears. Figure 2-4 shows the initialization window from the Report Writer.

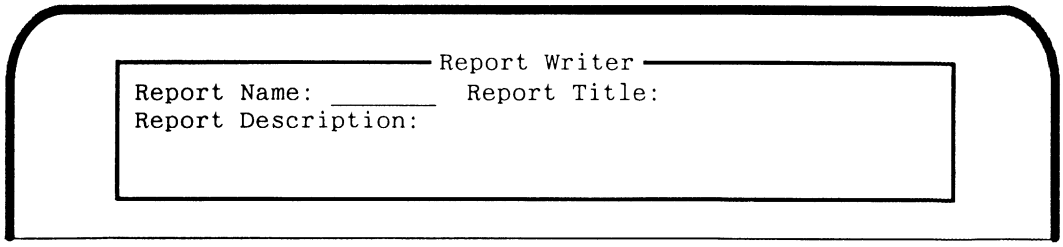


Figure 2-4: The Report Writer Initialization Window

FAST TRACK windows contain labeled fields. When FAST TRACK marks a field by underlining it (or highlighting it on DOS systems), this field is *active* — ready for your input. To fill in an active field, enter valid data and press either the [RETURN] key or the [GO] (F1) key.

When you press the [RETURN] key, FAST TRACK accepts the data you have entered in the active field. When you press the [GO] (F1) key, FAST TRACK accepts the values entered into all fields in the window. To edit values while still in an active field, use the [BACKSPACE], [DEL], [←] or [→] keys.

2.4 THE CHOICES WINDOW

If you are unsure what to type in a field, you can get a list of valid field entries by pressing [CHOICES] ([ESC] [C]). If no valid entries exist, FAST TRACK displays a message to this effect. If valid entries do exist, FAST TRACK displays the Choices window.

Figure 2-5 shows the Choices window for comparison operators available in the QBF Generator. In other circumstances, the Choices window may display a list of files, fields, forms, reports or QBFs.

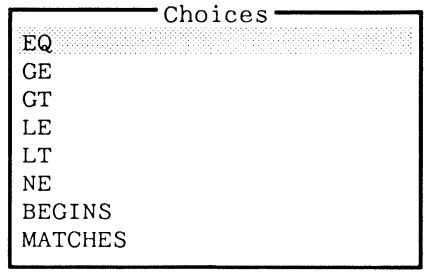


Figure 2-5: Comparison Operators

To select a value from the Choices window, use the following selection methods:

- Use the ↓ and ↑ keys to highlight a selection and press RETURN.
- Type the unique letter combination that corresponds to a selection and press RETURN.

In cases where you can only select one valid field entry, the Choices window disappears after you press RETURN. In cases where you can select more than one valid field entry, an asterisk (*) appears after each selected entry.

To deselect a choice, highlight the selection and press RETURN again. After you finish making selections, press the GO (F1) key to enter your selections and exit the Choices window. If you don't want to make a selection, or if no choices exist, press END (F4) to exit the Choices window.

2.5 MENUS

There are two types of menus in FAST TRACK: vertical menus and horizontal menus. These menu types appear in different locations and have different characteristics. The following sections describe each of these menu types in depth.

2.5.1 Vertical Menus

FAST TRACK uses vertical menus to display major program options. The Main Menu is an example of a vertical menu. It displays nine options as shown in Figure 2–6.

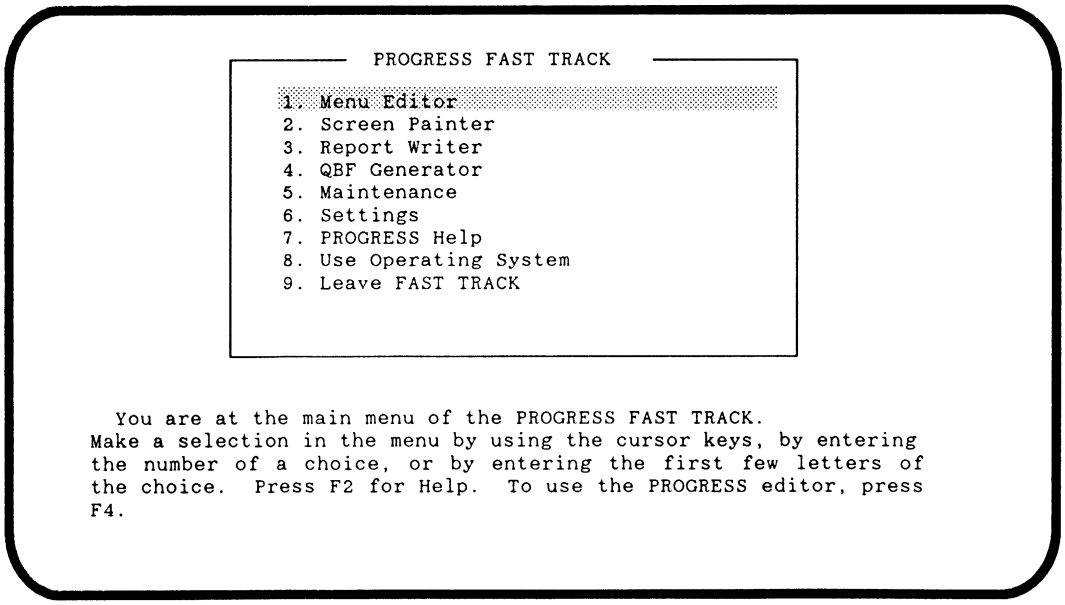


Figure 2–6: FAST TRACK Main Menu

There are three methods of selecting options from a FAST TRACK vertical menu:

- Use either the [↓] and [↑] keys, [SPACEBAR], or [TAB] to highlight an option. Then press [GO] (F1) or [RETURN].
- Type the first few letters of a menu option. FAST TRACK runs the option that corresponds to the first unique combination of letters that you type.
- Type the number of a numbered menu option.

2.5.2 Horizontal Menus

The horizontal command menu is the primary interface for the Menu Editor, Screen Painter and Report Writer modules. In each of these modules, the horizontal command menu appears at the bottom of the screen and is tailored to the module's needs.

FAST TRACK maintains a standard interface for the the horizontal menus in the Menu Editor, Screen Painter, and Report Writer — and whenever possible, it provides the same option naming conventions. Certain options such as `Help` and `Other` uniformly appear on all the horizontal menus.

To invoke a horizontal command menu, you must be in the appropriate editor and have previously entered data in the editor's initialization window. If this is the case, a message appears, prompting you to display the command menu. The default key sequence to display the menu is [OPTIONS] ([CTRL]-[O]).

After this process, FAST TRACK displays the menu across the bottom of your screen. Associated with each menu are three additional lines that give you information regarding menu options and screen status. FAST TRACK also highlights the currently selected option, which is the first option on the left of the menu. Figure 2-7 shows the horizontal menu from the Screen Painter editor.

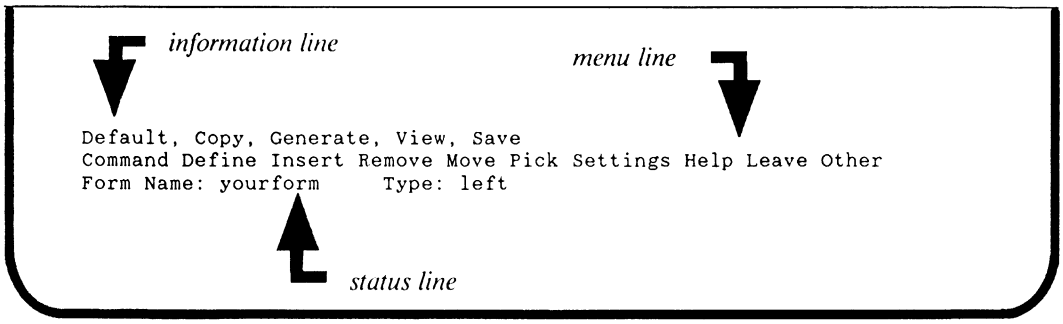


Figure 2-7: Typical Horizontal Menu

Horizontal command menus consist of three lines. They are:

- The *information line* — gives information about the function of the command that is currently highlighted.

- The *menu line* — shows the commands which execute an action or access another menu.
- The *status line* — displays the current form name and (in some editor functions) indicates the status of insert mode.

When necessary, FAST TRACK displays error messages and PROGRESS related messages (such as `Press Space bar to continue`) in the area beneath the horizontal command menu. Additionally, if you press a key not associated with a horizontal menu option, FAST TRACK displays an error message above the menu.

- Use the arrow keys, [SPACEBAR], or [TAB] to highlight your choice. Then press [GO] (F1) or [RETURN].
- Press the first letter of the menu choice. Horizontal menu options have unique first letters.

Many editing commands in the horizontal menus can also be invoked using key sequences or function keys. As explained in the section that describes the Settings window, FAST TRACK gives you the ability to customize these key settings. Using key sequences or function keys to move between editor commands can save you time and keystrokes.

Another fast way to invoke the horizontal menu options is to quickly type [CTRL]-[O], followed by the first letter of each option in the sequence of options that produces the desired command. For example, [CTRL]-[O]-[I]-[F] invokes the Insert-Field option in the Screen Painter without displaying the horizontal menus on UNIX systems.

To back up a level in the structure of a horizontal menu, use the [CLEAR] (F8) key. To exit a horizontal menu and return to the current editor, press the [END] (F4) key.

2.6 SHARED COMMANDS

The Menu Editor, Screen Painter and Report Writer share several commands that work identically in each of the three modules. The following sections describe these commands.

2.6.1 HELP

The HELP option provides help screens associated with the command menu of the current editor module. When you invoke the HELP option, FAST TRACK displays a screen tailored to the horizontal menu commands. For example, FAST TRACK displays the following screen when you invoke HELP in the Report Writer.

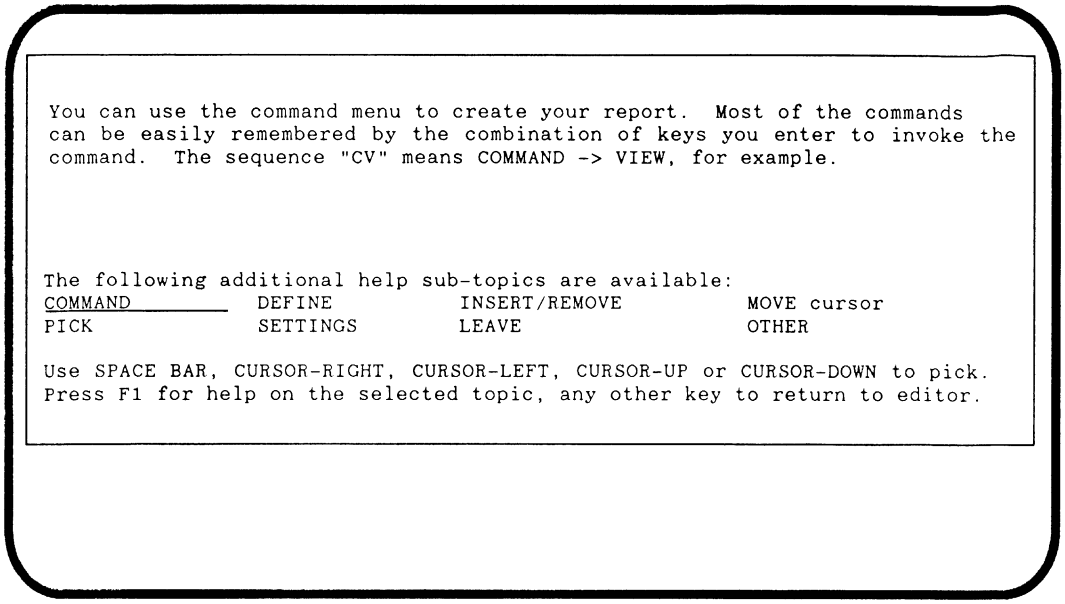


Figure 2-8: Help Window

To get help on a particular command option, use [SPACEBAR], or the cursor control keys to select a command option. On some systems, FAST TRACK denotes a selected command by underscoring it. On other systems, including most DOS systems, FAST TRACK highlights the command option.

After selecting the command option, press the [GO] (F1) key. To return to the editor module, press [SPACEBAR] or any of the alphanumeric keys.

2.6.2 LEAVE

The LEAVE command appears on the horizontal command menu of all the editor modules. In the Menu Editor, invoking the LEAVE command, causes FAST TRACK to immediately return to the Main Menu. This is possible because the Menu Editor saves your work after each menu item you create.

In the Screen Painter and Report Writer modules, the Leave command provides a two-item submenu: Save and QUIT. These options perform as their names indicate: SAVE stores your work to the default media, and QUIT causes FAST TRACK to return to the Main Menu without saving any work you have done since the last time you saved.

2.6.3 OTHER

The OTHER command option is identical in all three editor modules. It provides a submenu of system functions to assist you in controlling your FAST TRACK environment.

The following sections describe each command and the function associated with it.

OTHER→OPSYS. The OTHER→OPSYS command lets you leave FAST TRACK temporarily at any point to use your operating system utilities. To return to the same point where you left FAST TRACK, press CTRL–D if you are using UNIX, type **exit** if you are using DOS, type **PROGRESS Exit** if your are using BTOS/CTOS, or type **logout** if you are using VMS.

OTHER→DICTIONARY. The OTHER→DICTIONARY command allows you to use the PROGRESS Data Dictionary. For information on using the Data Dictionary, refer to the *PROGRESS Language Tutorial*.

OTHER→MAIN-MENU. You use the OTHER→MAIN-MENU command to return to the FAST TRACK Main Menu from any point in an editing session. If you are in the Menu Editor when you invoke this command, all of your changes are automatically saved. If you are in the Screen Painter or Report Writer, FAST TRACK allows you to save any changes before returning to the Main Menu.

OTHER→REPORTS. The OTHER→REPORTS command allows you access to a variety of development reports containing information about all of the FAST TRACK objects in your application. These reports are discussed in detail in Chapter 7 of this manual.

OTHER→GOTO. The OTHER→GOTO command lets you run FAST TRACK objects such as menus, QBFs, reports, or PROGRESS procedures. When you use this command, the given FAST TRACK object runs exactly as it appears in your completed application.

When you select OTHER→GOTO, FAST TRACK displays the window shown in Figure 2–9.

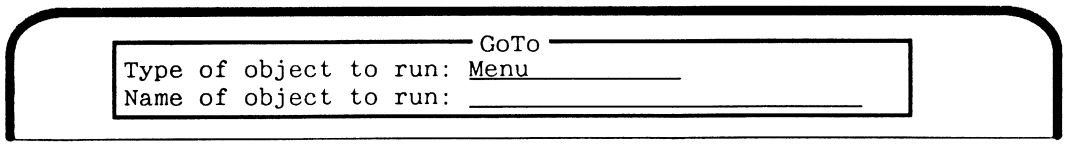


Figure 2–9: The GoTo Window

The following lists describes the different types of objects:

- Menu – Runs a menu. When you run a menu, you can test any action tied to your menu choices.
- QBF – Runs an already generated QBF.
- Procedure – Runs a PROGRESS procedure defined as a menu choice. If FAST TRACK cannot find the named menu on disk, it displays an error message informing you that the procedure does not exist.
- Report – Runs a report.

Chapter 3
The Menu Editor

The FAST TRACK Menu Editor gives you a quick way to build an end-user interface. All program flow — from running a simple PROGRESS procedure to exiting an application — can be controlled by a menu.

A FAST TRACK generated menu is a vertical menu, containing a stacked list of choices, or menu options. Each menu option is linked to a corresponding action such as running a FAST TRACK procedure. This chapter describes how the Menu Editor gives you the ability to quickly perform the following tasks:

- Create and edit menu options.
- Set menu display characteristics.
- Assign actions to menu options.
- Generate end-user applications.

3.1 BUILDING AN APPLICATION

The Menu Editor ties together the building blocks of your application. It lets you program on a menu-by-menu basis, or it lets you rapidly give a structure to FAST TRACK and PROGRESS procedures that you have already written and tested.

For large applications, the Menu Editor is best used when the building blocks already exist. It is good programming practice to write and test the majority of the procedures in an application before tying them together. Typically, you should plan your application, build procedures using FAST TRACK's Screen Painter, Report Writer, and QBF modules, and then use the Menu Editor.

On the other hand, the Menu Editor does not prevent you from creating menus before developing the rest of your application. In fact, if you link a newly created menu option to a non-existent Report Writer or QBF procedure, the Menu Editor transfers control to the appropriate module, in which you can create the report or QBF.

NOTE: When you attempt to link a menu option to a PROGRESS procedure, the Menu Editor attempts to run the procedure. If the procedure does not exist, FAST TRACK displays a message to notify you of the situation.

The Menu Editor has no limitations on the type of procedures or utilities it can invoke. You can use it to run FAST TRACK objects (menus, screens, reports, and QBFS); or you can *tie* a PROGRESS or FAST TRACK generated procedure to a menu item. Table 3-1 lists some typical as well as specialized examples.

Table 3-1: Some Menu Invoked Actions

Example	Purpose
Report	Call a FAST TRACK report module.
Data Dictionary	Access the PROGRESS Data Dictionary.
End-User Help	Provide PROGRESS Help in your application.
PROGRESS Procedure	Invoke any PROGRESS procedure.
Submenus	Call up to 9 additional levels of menus.
Utility Application	Run a utility application using operating system.
Exit	Leave the FAST TRACK application.

FAST TRACK automatically stores menu information as objects in database files. They include not only the information on the menu itself, but also information on any actions tied to a menu choice. As an alternative, however, you can create menu procedure (.p) files to store menu data. This method is typical for storing top-level menu information.

NOTE: When creating files, do not use the following filenames: REPORT, MENU, FT, FORM, MAINT, HELP, or QBF. These are names of FAST TRACK subdirectories. If you inadvertantly give a file one of these names, FAST TRACK will not operate properly.

3.2 THE TOP-LEVEL MENU

Each FAST TRACK application has a top-level menu, usually referred to as the main menu. The main menu is the first menu the user sees. Your application can have only one main menu, although you can create several top-level menus and selectively invoke them from an operating system script or batch file.

Typically, a main menu invokes one or more submenus, which invoke additional submenus or perform a function such as a database query or record update. In this manner, your system of menus ties all of the pieces of your application together and makes them accessible to the end-user.

To run your application as a standalone program (without accessing it through the FAST TRACK interface as you do during development), you must create a menu procedure (.p) file for the top-level menu (see section 3.9). As with all menus, FAST TRACK also stores the top-level menu as an object in a database file. Figure 3-1 shows the partial structure of a menu application.

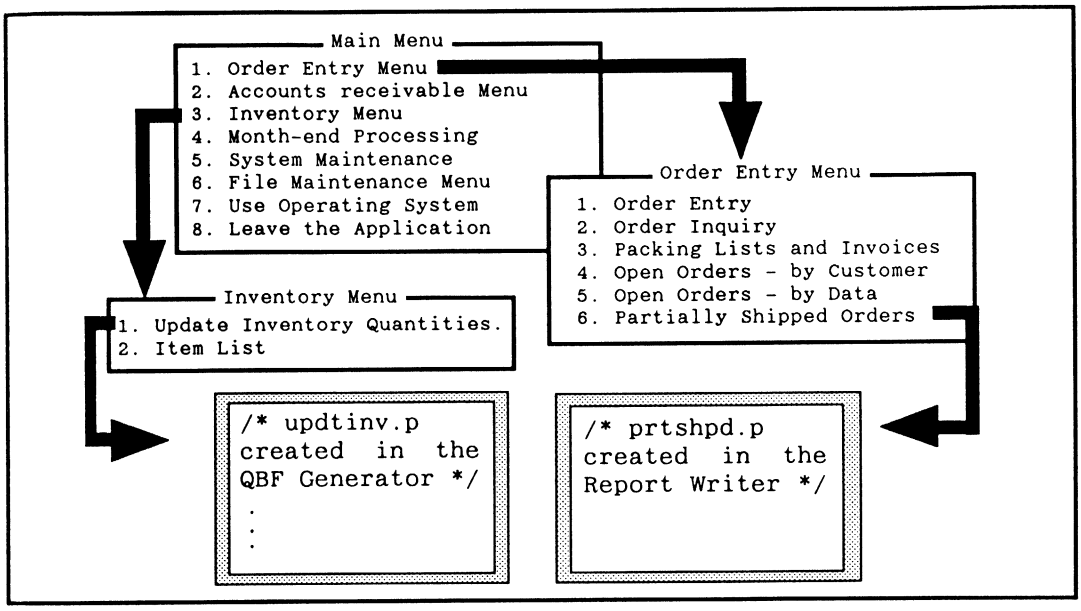


Figure 3-1: Example Menu Structure

In Figure 3-1, the top level menu is the main menu. It consists of eight choices. At the lowest level are FAST TRACK-generated procedure files: one created in the QBF Generator, and the other one created in the Report Writer. Note that both of these menu procedures have a .p extension; if you do not add the .p extension when you name a procedure, FAST TRACK automatically appends it to the procedure name.

Selecting an option from a menu invokes a corresponding action. For example, selecting option 1 from the Main Menu in Figure 3-1 displays the "Order Entry Menu." Similarly, selecting option 3 displays the "Inventory Menu." The other "Main Menu" options in Figure 3-1 invoke procedures that perform system tasks such as using the operating system and exiting the application.

At the lowest level, as seen in the "Partially Shipped Orders" option on the "Order Entry Menu," options invoke various actions. These include choices tied to a PROGRESS procedure, a QBF, an exit to the PROGRESS editor, a return to a previous menu, or possibly an exit to leave the application entirely. Menus created with the Menu Editor can have as many as 10 levels, including your main menu.

3.3 DEVELOPMENT PATH

To create a menu, you enter the Menu Editor, provide initialization data, and link menu options to program actions. The horizontal command menu described in Chapter 2 is your chief development tool in the Menu Editor. The following list provides a typical sequence of steps for creating a menu after you invoke the Menu Editor option from the FAST TRACK Main Menu:

1. Enter descriptive data in the Menu Editor initialization window.
2. Invoke the SETTINGS→MENU option on the command menu, if you want to change any Menu Editor defaults.
3. Enter menu options into the Menu Editor's main editing screen.
4. Assign program actions to menu options using the SETTINGS→CHOICE on the command menu.
5. Test your menu, using COMMAND→VIEW.
6. Generate a procedure file for your menu using COMMAND→GENERATE.
7. Exit from the Menu Editor, using the LEAVE command.

The Menu Editor also provides additional features through the command menu. Table 3-2 lists the command submenus for each command menu option. Refer to Appendix A for a complete description of all Menu Editor commands.

Table 3-2: Menu Editor Commands

Command Menu	Associated Submenu Options
Command	Copy List Generate View
Insert	One option: insert a menu line only.
Remove	One option: remove a menu line only.
Move	Next Previous Up Down
Settings	Menu Choice Insert/Overstrike
Help	Access PROGRESS Help.
Leave	Exit Menu Editor. Data already saved.
Other	Opsys Dictionary Main-Menu Reports Goto

3.4 STARTING THE MENU EDITOR

To enter the Menu Editor, select the Menu Editor option from the FAST TRACK Main Menu. When you do, FAST TRACK displays an initialization window as shown in Figure 3-2.

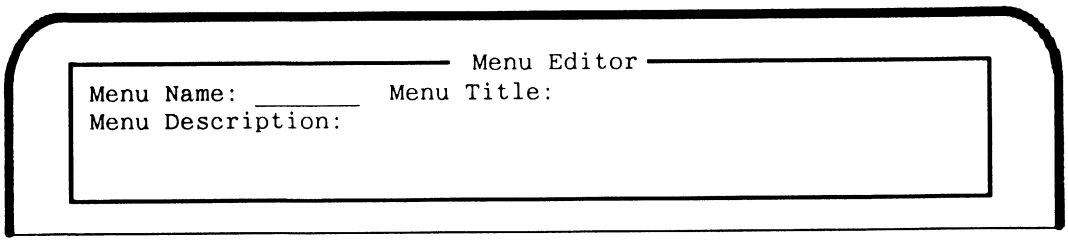


Figure 3-2: Menu Editor Initialization Window

You must enter a menu name before proceeding to the Menu Editor editing screen. Optionally, you can enter a title and description. The following descriptions explain the fields in the Menu Editor initialization window:

`Menu Name.` FAST TRACK uses the Menu Name as its key for storing information in its database. Additionally, if you do not specify a Menu Title, FAST TRACK automatically uses the Menu Name as the title.

`Menu Title.` Enter the title that you want associated with your menu. FAST TRACK automatically centers the Menu Title on the top line of the Menu Editor's editing screen.

`Menu Description.` If you want, you can enter a brief description of your menu in this field.

After you finish entering data into the Menu Editor initialization window, press RETURN or GO (F1). FAST TRACK displays the Menu Editor's main editing screen, which is depicted in Figure 3-3.

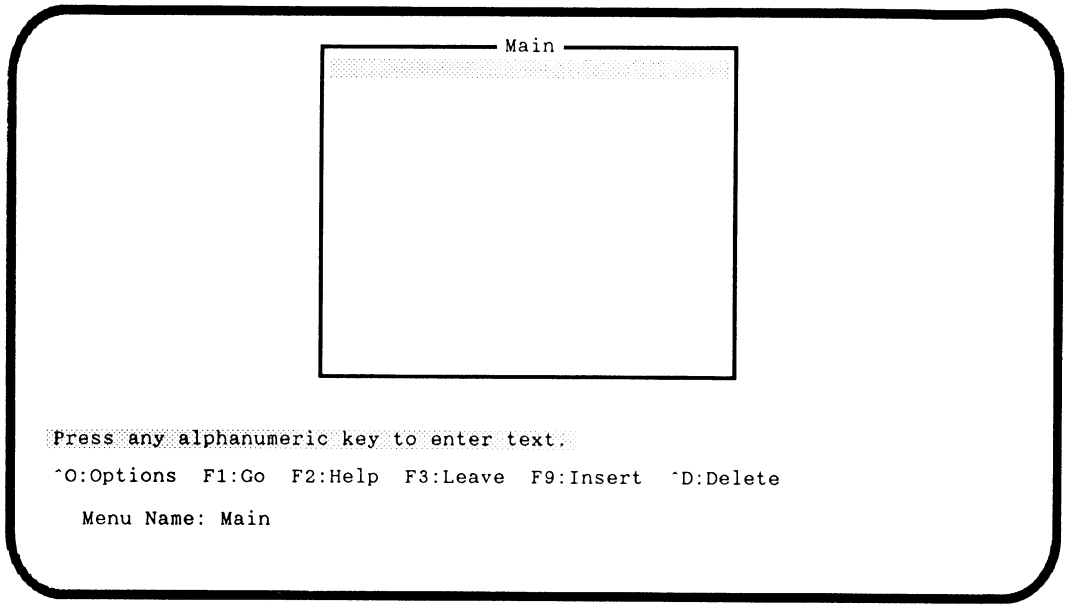


Figure 3–3: Menu Editor's Editing Screen

3.5 CHANGING DEFAULT SETTINGS

The SETTINGS→MENU option gives you a way to change the default settings for the menus you create. The default settings include display characteristics, text matching, and menu path selection, among other things. Figure 3–4 shows the Menu Settings window, which FAST TRACK displays when you invoke the SETTINGS→MENU option.

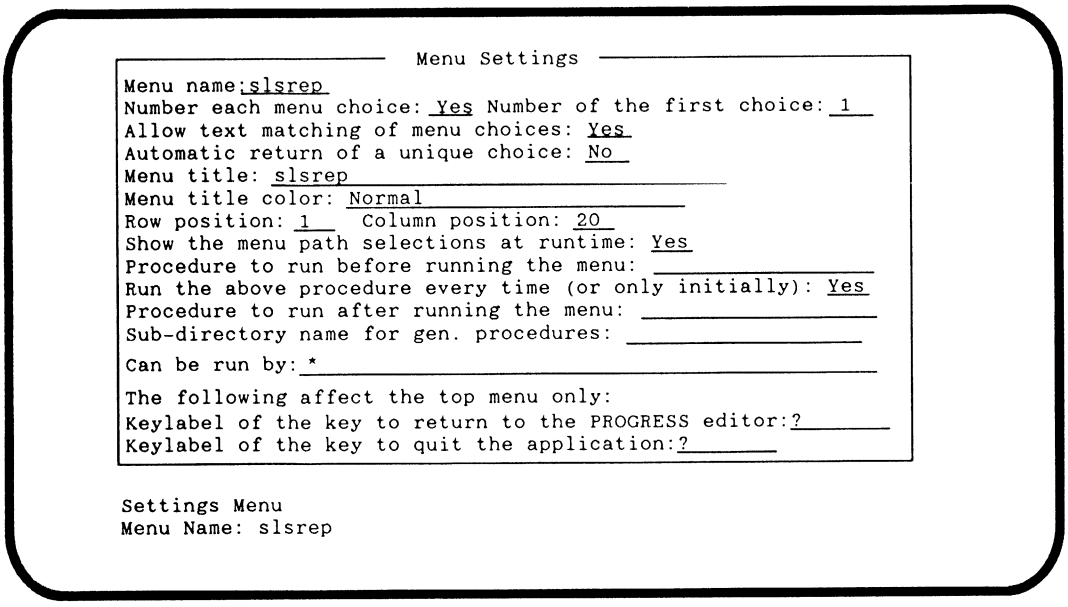


Figure 3–4: Menu Settings Window

To edit a default setting, type the appropriate text in the field next to the setting label. To move between fields, use any of the cursor movement keys described in Chapter 2. When you have finished editing the settings, leave the window and save your changes by pressing GO (F1).

You can access the Menu Settings at any point in the development path, but generally, you should determine most of your settings before you begin creating menus. The following sections explain the effects of each setting.

`Menu Name.` The menu name that you entered in the Menu Editor's initialization window is displayed here.

`Number each menu choice.` Menu options are numbered by default. Enter **No** to inhibit the numbering of menu options. If you do number the menu options, the end-user can select an option by typing its number.

`Number of the first choice.` By default, menu option numbering begins with the number 1. Change this setting to begin numbering at a number other than 1. The starting number must fall in the range of 0 to 975. You can enter a maximum of 25 choices per menu.

`Allow text matching of menu choices.` If you specify **Yes** to this setting, the end-user can select a menu option by entering its first unique sequence of letters. For example, if a menu had two options named "Return" and "Replace," typing *ret* would invoke the former and *rep* the latter. The default for this setting is Yes.

Automatic return of a unique choice. This setting determines whether the action tied to a menu option is invoked when the end-user selects it. The default for this setting is **No**, which requires the end-user to press RETURN after selecting an option.

Menu title. By default, the menu name is used as the title. Modify this setting to change the menu title. The menu title can be up to 40 characters in length. You can enter letters, numbers, spaces and most keyboard characters into the title.

Menu title color. Your menu title can appear with normal screen attributes, in reverse video, underlined, blinking, or in any color available on your monitor. The colors and color attributes available depend on the monitor on which you intend to run your application. You can use the following settings as the title color on any monitor:

- **NORMAL.** Green or amber on a black background.
- **MESSAGES.** Reverse video.
- **INPUT.** Underlined.

The default is **NORMAL.** For additional information on color settings, refer to the "Color Phrase" in the *PROGRESS Language Reference* manual.

Row Position. The Row Position setting determines the upper left horizontal coordinate of your menu. The default setting is 1.

Column Position. The Column Position setting determines the upper left vertical coordinate of your menu. The default setting is 1.

Show the menu path selections at runtime. If this setting is **Yes**, FAST TRACK displays the menu titles that the end-user has invoked to get to the current location in the application. Additionally, as the end-user backs through a menu path, FAST TRACK removes menu titles from the menu path display.

Procedure to run before running this menu. This setting lets you run a PROGRESS procedure before FAST TRACK invokes the specified menu. The procedure must be located in a directory accessible to your PROPATH environment variable, or you must specify the full path and filename here.

Run the above procedure every time (or only initially). If you want the procedure specified by the previous setting run every time the end-user selects the menu, enter **Yes**. Enter **No** if you want the procedure to run only the first time the end-user selects the menu. The default value is Yes.

Procedure to run after running the menu. This setting lets you run a PROGRESS procedure after FAST TRACK invokes the specified menu. The procedure must be located in a directory accessible to your PROPATH environment variable, or you must specify the full path and filename here.

`Subdirectory name for gen. procedures.` This setting specifies the subdirectory where you want FAST TRACK to store your menu procedure files.

`Can be run by.` This setting limits access to the menu. The default is an asterisk (*), which allows all users to access to the menu. For additional information on user authorization, refer to Chapter 7 of this manual or the CAN-DO function in the *PROGRESS Language Reference* manual.

`Keylabel of the key to return to the PROGRESS editor.` This setting lets you specify a special key to let the end-user escape to the PROGRESS editor from the top-level menu of the application. For example, if you run the application with the -p parameter, which prevents access to the PROGRESS editor, enter a keylabel in this field to provide access to the editor. (See the *Programming Handbook* for more information about start-up options and keylabels.)

`Keylabel of the key to quit the application.` This setting lets you define a special key that exits the application. (See the *Programming Handbook* for a list of keylabels.)

3.6 CREATING MENU OPTIONS

To create a menu option, which is also referred to as a *Choice Label*, enter text directly into the Menu Editor editing screen. This window appears after you have completed the initialization window, and optionally, modified the menu's default settings.

FAST TRACK automatically places the new title at the top of the Menu Editor window. FAST TRACK also numbers options as you create them, unless you previously changed the setting. Figure 3-5 shows a partially created "Main Menu" that uses the automatic numbering feature.

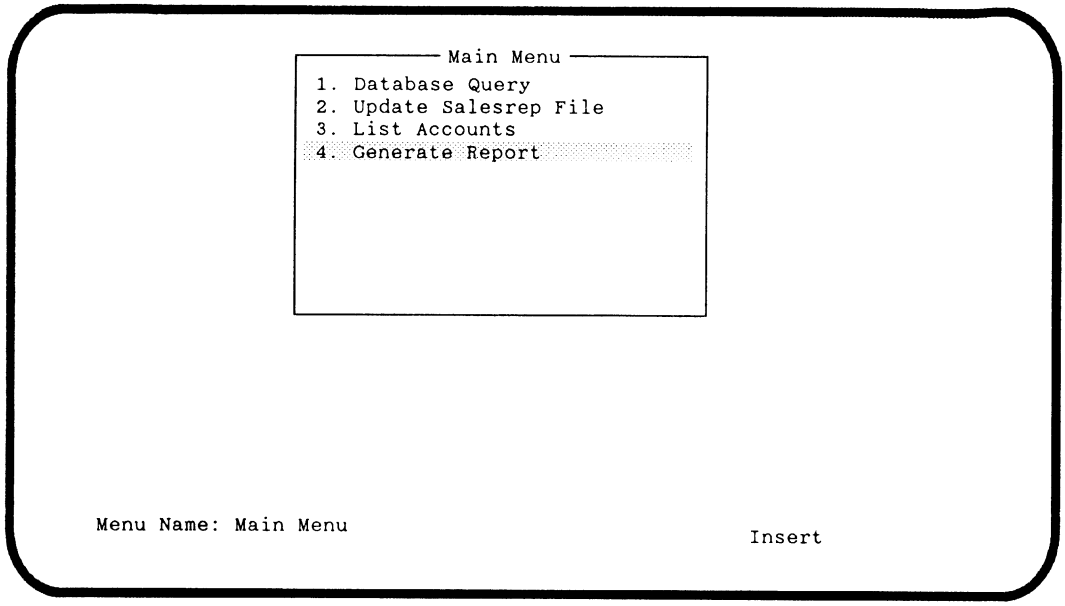


Figure 3-5: Partially Created Menu

You can enter up to 25 options (choices) per menu. As soon as you enter all 25 options, FAST TRACK returns you to the first item. Be concise when naming options. You can enter up to 40 characters per option, but if you choose to have FAST TRACK number each option, only the first 35 characters are displayed. An option cannot span more than one line.

When you finish entering an option, press RETURN. This advances the cursor to the next line, where you can enter another option name. Additionally, you can use the INSERT, REMOVE, and MOVE commands to edit an existing menu. Refer to Appendix A for a complete description of these commands.

3.7 ASSIGNING MENU ACTIONS

To assign a menu action, invoke the SETTINGS→CHOICE command. This command displays the Menu Choice Settings window depicted in Figure 3-6.

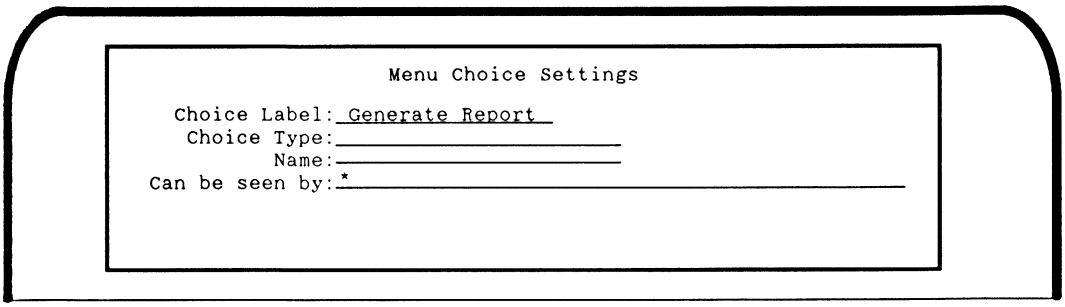


Figure 3-6: Menu Choice Settings Window

The Menu Choice Settings window always displays the menu option highlighted in the Menu Editor window. The option name appears in the Choice Label field. You can change the option name from within this window, or you can change from the Menu Editor editing screen.

Before you designate a specific action for the menu option, you must define an action type in the Choice Type field. To do this, enter the appropriate text or press CHOICES (ESC C) to get a list of valid types. Next, enter an action name in the Name field and then, in the last field, enter any user restrictions that you require. The following sections further describe the fields in the Menu Choice Settings window.

`Choice Type`. Any of the following object types or actions may be entered into this field.

- **Menu** - Invokes the submenu specified in the Name field.
- **QBF** - Runs the QBF specified in the Name field.
- **Quit** - Exits the application to the operating system.
- **Procedure** - Runs the PROGRESS procedure specified in the Name field.
- **Report** - Runs the report specified in the Name field.
- **Return** - Returns control to a previous menu.
- **Text** - There are no actions associated with this choice. It allows you to include text or blank lines in the menu.

NOTE: If you specify **Return** as the Choice Type in the top-level menu of an application, the menu option invokes the PROGRESS editor. On a system with the PROGRESS FAST TRACK Run-time Utilities, specifying **Return** from a top-level menu causes the application to exit to the operating system.

`Name`. You must enter the name of the menu, QBF, PROGRESS procedure, or report that you want to assign as the menu choice action. FAST TRACK does not prompt you for this field if you have chosen Quit or Return as the choice type. Use CHOICES (ESC C) to see a list of available choices. The name you enter cannot have more than 40 characters.

`Can be seen by`. You define access privileges for each menu option. This field determines which end-users can access the given menu option. The default setting is an asterisk *, which allows all end-users to access the option. For more information on security, refer to Chapter 7 of this manual.

3.8 CREATING A SUBMENU

To create a submenu, define the choice type as **menu**, give it a name and specify any access privileges. When you press GO (F1), the Menu Editor tries to display the submenu. If no submenu exists, it displays a blank menu and lets you create the submenu. You edit the submenu and assign it menu options exactly as you do for a main menu.

NOTE: You can only have 10 levels of menu, including your main menu.

3.9 MENU FILES

When you have completed your menu, you can make a menu procedure (**.p**) file in which to store the related code. Typically, a menu procedure file is necessary only if you are creating a main menu, or several start-up menus to invoke from an operating system script or batch file. Otherwise, all menus in your application are maintained as objects in the FAST TRACK database.

If you want a top-level menu to be the entry point into your application, create a menu procedure file using the `COMMAND→GENERATE` option. You can use the resultant **.p** file as the start-up procedure for you application. Alternatively, you can invoke this procedure file from the PROGRESS editor or from another menu in your application.

To use a menu procedure file as a start-up procedure for an application, create a start-up script or batch file. The start-up procedure option for PROGRESS is specified by the -p parameter on UNIX, DOS, and BTOS/CTOS systems; and the /start-up parameter on VMS systems. Refer to the *Programming Handbook* for additional information on start-up parameters.

You may also want to create menu procedure files to test a menu from the PROGRESS editor, although you can test menus more quickly using the `COMMAND→VIEW` option.

The command options also provide tools for reviewing and copying your FAST TRACK menus. The following sections explain the command options. You can also refer to Appendix A for a summary of these options.

3.10 GENERATING A MENU FILE

Use the `COMMAND→GENERATE` command to generate a PROGRESS procedure file for a menu. When you choose this command, FAST TRACK prompts you for the name you want to give to the PROGRESS file. Figure 3-7 shows the COMMAND→GENERATE window.

```
Enter a name for the output PROGRESS file
Command Generate>  filename.p
Menu Name: main
```

Figure 3–7: Generate PROGRESS Code for the Application

After you enter a filename, press GO (F1) or RETURN. FAST TRACK generates and saves two files: *filename*.p, which is a procedure file; and *filename*.r, which is the compiled version of *filename*.p. The content of *filename*.p is the following:

```
                                                    filename.p

  {menu/mxmast.i "main"}
```

In the above line of code, "main" is the name of the FAST TRACK menu. Data for this menu is stored in the FAST TRACK database. Whenever you run a menu procedure, FAST TRACK reads this menu data from the database, then builds the associated items and displays them.

To create a start-up script or batch file, use the PROGRESS editor to make a copy of the `proft` start-up file. Once you have copied `proft`, rename the copied version and replace the reference to `ftcheck.p` with the name of your start-up procedure. The following line resembles the last line of the `proft` file on UNIX and DOS systems. Replaceable information appears in brackets:

```
set -e 55 -l 30 -c 30 -p <ftcheck.p> <dbname>
```

For additional information on start-up files, refer to the *Programming Handbook*.

3.11 TESTING A MENU

There are three ways to test menus: by using the COMMAND→VIEW option, which immediately displays the menu; by creating a menu procedure file with the COMMAND→GENERATE option and then running the file from the PROGRESS editor; and by using the GOTO option from the Screen Painter or Report Writer modules.

After you create a menu, you can immediately test it by using the COMMAND→VIEW option. Simply press CTRL-O and choose COMMAND and then VIEW. FAST TRACK displays the menu as it appears in your final application.

If you have saved your menu as a procedure file, you can test it from the PROGRESS editor. Simply enter the editor, call up the file, and execute it. You can also use the PROGRESS RUN statement to test a menu procedure file.

When you are working in another module such as the Report Writer, you can test a menu by using the GOTO command.

3.12 LISTING A MENU TREE

When you select this command, FAST TRACK displays an indented listing that outlines the current state of your menu system. This listing shows each menu option for each menu in your system. It also shows the type of action associated with each option and whether an a choice type has been defined for each menu option.

```
Menu Titles and Labels [Name]

Main
     Database Query [Undefined choice type]
    Update a Sales rep File [Menu slsrep]
        Update a sales Rep [Undefined choice type]
        Add a Sales Rep [Undefined choice type]
        Delete a Sales Rep [Undefined choice type]
        Return [Undefined choice type]
    Create a Sales Rep Report
    Use Operating System [Procedure tstos.p]
    Exit [Undefined choice type]
```

Figure 3-8: Outline of Main Menu

3.13 COPYING A MENU FILE

Use the COMMAND→COPY command to copy a menu. When you make this selection, FAST TRACK prompts you for the name of the menu to copy.

After you enter the menu name and press GO (F1) or RETURN, FAST TRACK overwrites the old menu information with the contents of the copied menu file. The title of the copied menu appears in the top border of the frame. Note, however, that the name of your menu has not changed.

3.14 OTHER FAST TRACK MODULES

When you specify `Report` or `QBF` as a a choice type and run that choice with the [GO] (F1) key, FAST TRACK invokes the corresponding editor. For example, if you specify `Report` as the choice type, FAST TRACK invokes the Report Writer. If you have previously created the report or QBF, FAST TRACK loads it. If no report or QBF exists, FAST TRACK gives you the opportunity to create one by invoking either the Report Writer or QBF module.

FAST TRACK also prevents you from accidentally quitting your development environment. If you invoke a menu option that quits to the operating system, FAST TRACK displays the message, "The user will QUIT (RETURN) now," then prompts you to continue.

Chapter 4
The Screen Painter

The Screen Painter lets you design and create *wysiwyg* (what-you-see-is-what-you-get) forms for your applications. You design forms by entering database fields, labels, and associated text directly in the Screen Painter editing screen. Because the Screen Painter creates forms from the data you enter, you know that your design is going to work once you have created it.

The Screen Painter permits you to define data entry fields as standard PROGRESS data types—character, date, decimal, integer, or logical. After you have finished a form, you can generate a PROGRESS form procedure, which is stored in a file with a `.f` extension. You can then include this file in a PROGRESS procedure for use with the `DISPLAY`, `UPDATE`, `SET` and `PROMPT-FOR` statements.

This chapter introduces the following subjects:

- Development path for creating customer forms using the Screen Painter..
- Starting the Screen Painter.
- Changing default settings of screens and fields to give reports a custom appearance.
- Creating a form using the Screen Painter.
- Editing a form using the Screen Painter.
- Defining form files so that their fields may be inserted.
- Defining variable fields such as: a string of characters, the date, a decimal, an integer, or a logical.
- Generating a form file which can be used in a PROGRESS procedure.
- Generating a QBF using a form produced by the Screen Painter. This is practical when you need to define a single file QBF. For additional information on defining QBFs, refer to Chapter 6 of this manual.

4.1 DEVELOPMENT PATH

The Screen Painter offers a straightforward development path for creating custom forms. The horizontal command menu described in Chapter 2 is your chief development tool in the Screen Painter.

The following list provides a typical sequence of steps for creating a custom form after you invoke the Screen Painter option from the FAST TRACK Main Menu:

Enter descriptive data in the Screen Painter initialization window.

If necessary, use the SETTINGS→SCREEN and SETTINGS→FIELD commands to change the default form settings.

Compose descriptive information for a form by entering text directly into the Screen Painter editing screen.

If you want all form fields in a single file, use the COMMAND→DEFAULT option to generate the form. Alternatively, invoke the DEFINE→FILE option to define the files used in generating the form. (If you use the form to generate a QBF, you can define only one file. If you intend to access the form from a PROGRESS procedure, you can define up to 12 files.)

Use the INSERT submenu options to enter labels, field, and data in the Screen Painter editing screen. (You can also insert variables in a field, but you cannot generate a QBF for a screen containing a variable.)

Test your form, using the COMMAND→VIEW option.

Generate a file for your form using the COMMAND→GENERATE option. (You can later invoke this form from a PROGRESS procedure.)

Exit from the Screen Painter, using the LEAVE option.

NOTE: The Screen Painter displays an error message when you attempt to insert a field that is wider than the current frame width of the screen.

The Screen Painter also provides other features through the command menu. Table 4–1 lists the command submenus for each command menu option. Refer to Appendix B for a complete description of all Screen Painter commands.

Table 4–1: Screen Painter Commands

Command	Submenu Options
COMMAND	Default Copy Generate View Save
DEFINE	File QBF
INSERT	Field Label Data Row Column
REMOVE	Field Row Column Picked
MOVE	Next Previous First End Top Bottom Up Down Left Right
PICK	Object Field Area Move Undo
SETTINGS	Screen (Form) Field Mode
HELP	Access help information for various commands
LEAVE	Save Quit
OTHER	Opsys Dictionary Main-Menu Reports Goto

4.2 STARTING THE SCREEN PAINTER

To enter the Screen Painter, select the Screen Painter option from the FAST TRACK Main Menu. When you do, FAST TRACK displays the initialization window shown in Figure 4–1.

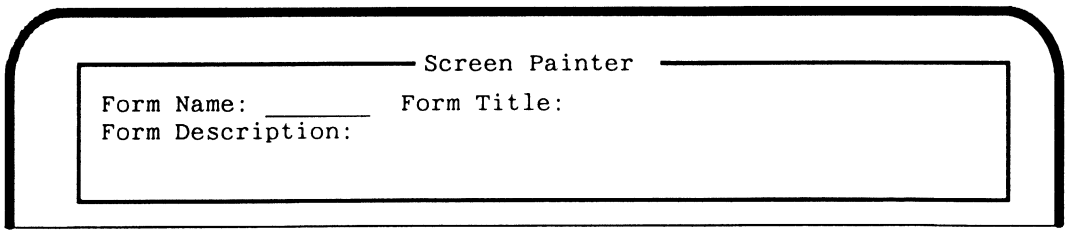


Figure 4–1: The Screen Painter Window

This window requests three pieces of information: Form Name, Form Title, and Form Description. The following sections describe these fields.

`Form Name`. You must enter a form name. It can be up to seven characters long and only contain alphanumeric characters and the underscore symbol.

`Form Title`. The form title is optional. If supplied, the form title appears centered in the top border of the form when you view the form on the screen.

`Form Description`. The Form Description is optional. It is for documentation purposes only, and can contain any information that you want to convey about the purpose of the form.

After you complete the initialization window, press [GO] (F1). FAST TRACK displays the Screen Painter editing screen, in which you create your custom form. At this point, you can begin creating your form, or change the form's default screen and field settings, which are explained in the next two sections.

NOTE: Changing a Data Dictionary field definition does not change the definition of that field in an existing Screen Painter form, or your `.f` file.

4.3 CHANGING DEFAULT SETTINGS

The `SETTINGS` options let you define the characteristics of your forms as well as the fields in your form. The `SETTINGS→SCREEN` command changes the window characteristics of the form. The `SETTINGS→FIELD` command displays the Field Attribute Setting window, which allows you to change default field characteristics. The next two sections explains these commands.

4.3.1 Screen Settings

You can give your application a custom appearance by changing the default characteristics of the screens (frames) that you create. Use the `SETTINGS→SCREEN` command to invoke the Screen Settings window. Figure 4–2 shows an example Screen Settings window.

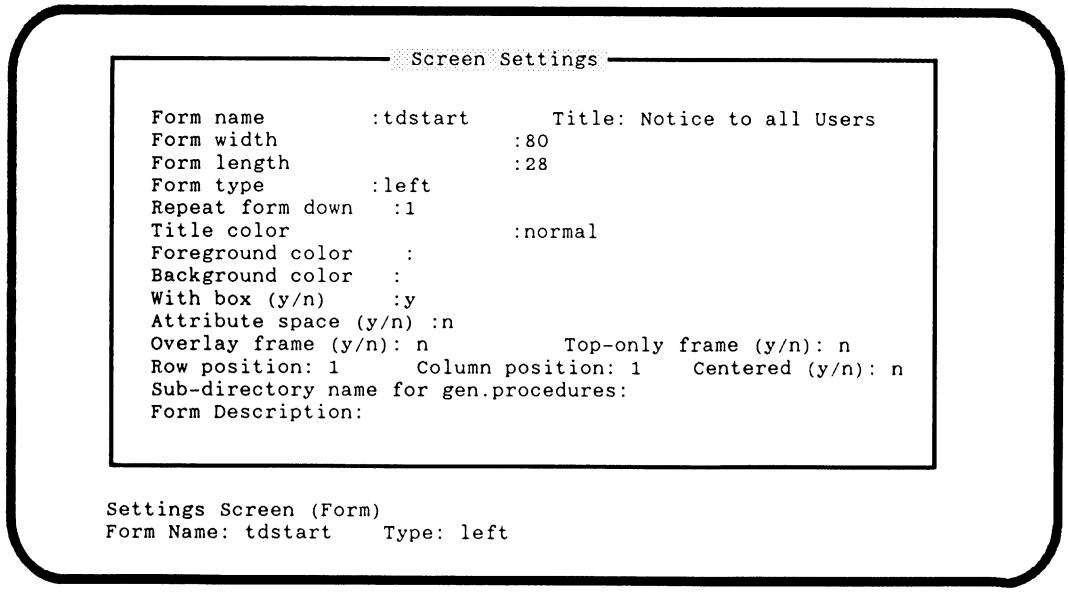


Figure 4–2: The Settings Screen Window

To edit a default setting, type the appropriate text in the field next to the setting label. To move between fields, use any of the cursor movement keys described in Chapter 2. When you have finished editing the settings, leave the window and save your changes by pressing GO (F1).

You can access the Screen Settings window at any point in the development path, but generally, you should determine most of your settings before you begin creating forms. The following sections explain the screen settings.

Form name. Specifies the name by which the procedures in your application refer to the form. This is the form name that you provided in the initial Screen Painter window.

Title. Contains the title that appears at the top of the form. This item is optional. You can enter a title only if you also enter **Y** for yes in the With box field, which is explained later.

Form width. Specifies the number of characters that can fit on one line of the form. The maximum width is 255 characters. Note that you cannot display a form that is wider than 80 columns on your screen, unless you are using a special video display. The extra width is meant primarily for forms that are sent to the printer or a file. The default width is 80 characters.

Form length. Specifies the number of lines that the form requires. The form can be up to 66 lines long. The user can scroll forward and backward through forms that are longer than the screen. The default length is six lines less than the number of lines that your terminal supports. To view the form from the Screen Painter, ensure the form does not exceed the number of lines that your terminal supports.

Form type. Determines where labels appear in relation to the data area of the field. The form type can have one of the following values:

- **left** - Labels appear to the left of the data.
- **top** - Labels appear above the data.
- **header** - Indicates that this is a header frame. Refer to the "DISPLAY" statement in the *PROGRESS Language Reference* manual if you want to edit the form file and modify the header frame.

The default form type is *left*.

Repeat form down. Determines how many records are displayed at one time with this form. The default value is 1. If you specify a value other than the default value, you will have scrolling capabilities when you generate a QBF against the form.

Title color. Sets terminal colors and attributes. Your menu title can appear with normal screen attributes, in reverse video, underlined, blinking, or in any color available on your terminal. The colors and color attributes available depend on the terminal on which you intend to run your application. You can use the following settings as the title color on any terminal:

- **normal** - green or amber on a black background.
- **messages** - reverse video.
- **input** - underlined.

The default is **normal.** For additional information on color settings, refer to the "Color Phrase" in the *PROGRESS Language Reference* manual.

Foreground color. Specifies the foreground color of the form. For other colors you can use, refer to the "Color Phrase" in the *PROGRESS Language Reference* manual.

Background color. Sets background color, which can be reverse video, underlined, blinking, or any color available on your terminal. For other colors you can use, refer to the "Color Phrase" in the *PROGRESS Language Reference* manual.

With box (Y/N). Sets border characteristics. Forms can be displayed on the screen with or without borders. You enter **Y** (for yes) if you want a box, or **No** (for no) if you do not. Forms cannot be *printed* with boxes. When forms with boxes are printed, the application leaves blank lines and columns where the box normally appears. Your form must display a box if it is to have a title.

Attribute space. Tells FAST TRACK whether or not you are using a spacetaking terminal. The default value is **N.** For more information about spacetaking and non-spacetaking terminals, see Chapter 7 in the *Programming Handbook.*

Overlay frame. This type of frame can overlay any other frame except a top-only frame (see below). The default value is **N.**

Top-only frame. Specifies frame layering. If you specify **Y** for this option, your frame cannot be covered by another frame. The default is **N.**

Row position. Determines the row in which the upper left corner of the form is located. The default value is 1.

Column position. Determines the column in which the upper left corner of the form is located. The default value is 1.

Centered. Specifies centering. If you specify **Y** for this option, the frame is anchored in the top row and centered between the left and right sides of the screen. The default value is **N.** This setting overrides the `Column position` setting. Note that this setting does not center actual data.

Sub-directory name for gen. procedures. Specifies a directory other than your current working directory, in which to save the PROGRESS procedure that contains your frame format statement (see section 4.8).

Form description. Modifies the form description that you entered in the Screen Painter initialization window.

After you change the Screen Settings values, press GO (F1) to make the new settings take effect. To resume editing without changing the settings, press END (F4).

4.3.2 Field Attribute Settings

You can give your application a custom appearance by changing the default field characteristics of a form. Use the SETTINGS→FIELD command to invoke the Field Attribute Setting window. Figure 4-3 shows an example Field Attribute Settings window.

```
Name:
Addr:
City:
State:
Zip:
Tel num:

              Field Attribute Setting

Field : Phone
File  : customer
Db    : ftdb
Format: (999) 999-999          Display only (y/n): n
Label : Tel num:
Update order: 50   Auto-return (y/n): n  Attr-space (y/n): n
Help  :
Validate: ?
Invalid Msg:

Press any alphanumeric key to enter text.
^O:Options  F2:Help  F4:Leave  F9:Insert  ^D:Delete
Form Name:  tstart ftdb       Type: left
```

Figure 4-3: The Field Attribute Setting Window

To edit a default setting, type the appropriate text in the field next to the setting label. You can only edit the underlined or highlighted fields. To move between fields, use any of the cursor movement keys described in Chapter 2. When you have finished editing, leave the window and save your changes by pressing GO (F1). The following sections explain the field attribute settings:

Field. The name of the field. You cannot change this setting.

File. The database file with which the field is associated. You cannot change this setting.

Db. The name of the database that the file is in.

Format. The data entry format. You can modify the default data format any way you like. Any modifications made to the data format in this window do not affect the data format in the Data Dictionary. See the description of the DEFINE→VARIABLE option in this chapter and Chapter 2 of the *Programming Handbook* for more information on data formats.

Display only. f you specify **Y** (for Yes) in this field, you can display the field, but cannot update it with this form when you use the form in a QBF. The default value is **N**.

Label. The default value is the label that is specified for the field in the Data Dictionary. You can override the default label by entering another label. When you override the default label, the new label does not affect the Data Dictionary.

Update order. The *update order* is the order in which you want the fields on your form updated when your form is used in a QBF. You enter an integer between 0 and 9,999. The default value is assigned in multiples of 10, based on the order in which the fields were inserted in the form. The update order must be unique. If you do not specify an order, fields are updated according to the order in which you inserted them.

Auto-return. If you specify **Y** as the value of this field, the user does not have to press RETURN after typing the last character in an update field. The default value is **N**.

Attr-space. Enter **Y** if you are designing the form for a spacetaking terminal. The default value depends on the the terminal you are using. Refer to Chapter 7 of the *Programming Handbook* for additional information on spacetaking terminals.

Help. The help message that you want to display at the bottom of the screen. Default messages are defined in the Data Dictionary.

Validate. Validates your output when you use the form in a PROGRESS SET, UPDATE, or PROMPT-FOR statement. The following is a typical validation:

```
max-credit >= 0 and max-credit <= 999999.99
```

In this example, the user will be prohibited from entering a number that is either negative or greater than 999,999.99.

The default value is ? (unknown) for no validation statement.

You will find information on validation in Chapter 3 of the *Programming Handbook.*

`Invalid Msg.` Specifies the message that appears when the end-user enters data that does not satisfy the Validate expression. For example:

```
Max credit must be between 0 and 999,999.99. Try again.
```

4.4 CREATING A FORM

Use the COMMAND→DEFAULT option to create a form with all of the fields in a single file. When you enter this command, FAST TRACK displays the window shown in Figure 4–4.

```
Enter database file name.  Press ESC-C for choices.
Command Default >_____________________________________
Form Name: test1        Type: left
```

Figure 4–4: The Default Form Prompt

If you cannot recall the filename you want to use, press CHOICES (ESC C) to see a menu of available files.

After you enter the name of the file for which you want to create a default form and press RETURN or GO (F1), FAST TRACK generates a form using the default form settings. This form is different than the PROGRESS default form. The FAST TRACK default form uses side labels. Information is displayed in two columns, rather than with top labels wrapping. If any field is wider than the frame width, FAST TRACK displays an error message and skips the field. FAST TRACK also skips any fields that do not fit into the frame.

You can edit this form after creating it. FAST TRACK uses the values for the settings available at the time that the COMMAND→DEFAULT command is selected. Therefore, if you want the form to have any characteristics other than the standard FAST TRACK default settings. Change them using the SETTINGS→SCREEN and SETTINGS→FIELD commands before selecting the COMMAND→DEFAULT option.

FAST TRACK enters all of the fields in the form as shown in the following example for the salesrep file in the demonstration database.

```
    Sales Rep: ____
         Name: _________________
       Region: __________
        Title: _______________
Yearly Quota: ___________
  Date hired: ___________




 Press any alphanumeric key to enter text.
 ^O:Options  F2:Help  F4:Leave  F9:Insert  ^D:Delete
 Form Name: srpform      Type: left
```

Figure 4–5: A Default Form

4.5 EDITING A FORM

You can type any text you need in your form directly into the Screen Painter editing screen. When you select COMMAND→VIEW or display a form in your application, the text appears exactly as you typed it.

You can use text to create banners, messages to users, explanatory references, or labels. The following figure shows an example of a form that consists only of text.

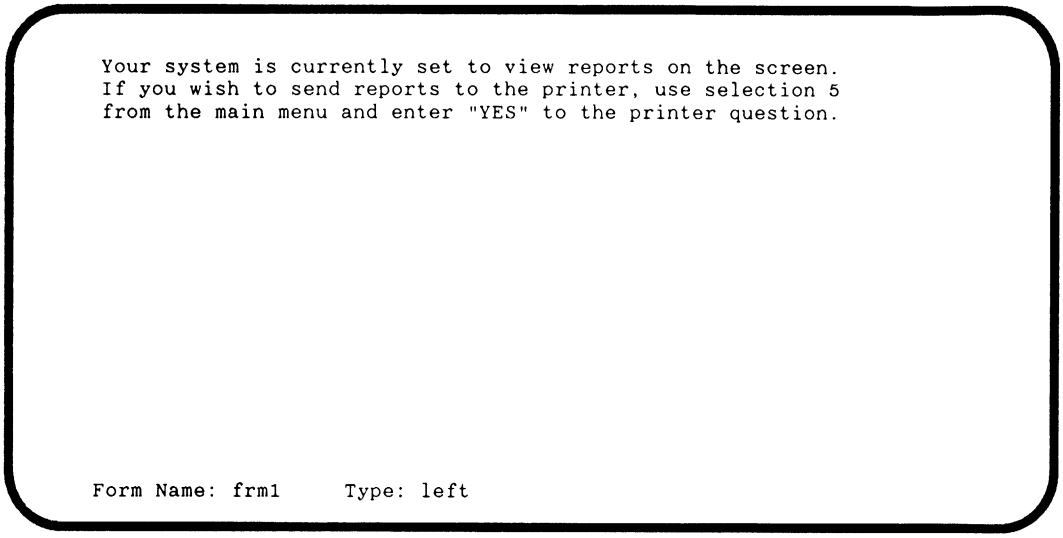


Figure 4–6: The Screen Painter Editor

In your application, the text in your form can appear in a screen location different from the location where you created it. Do not be alarmed. The text components of your form remain correctly positioned, relative to one another. You can control the final layout of your form by using the SETTINGS options (see section 4.3). Figure 4–7 shows the text from the previous example displayed as a centered box, using both the Centered and With box options from the Screen Settings window.

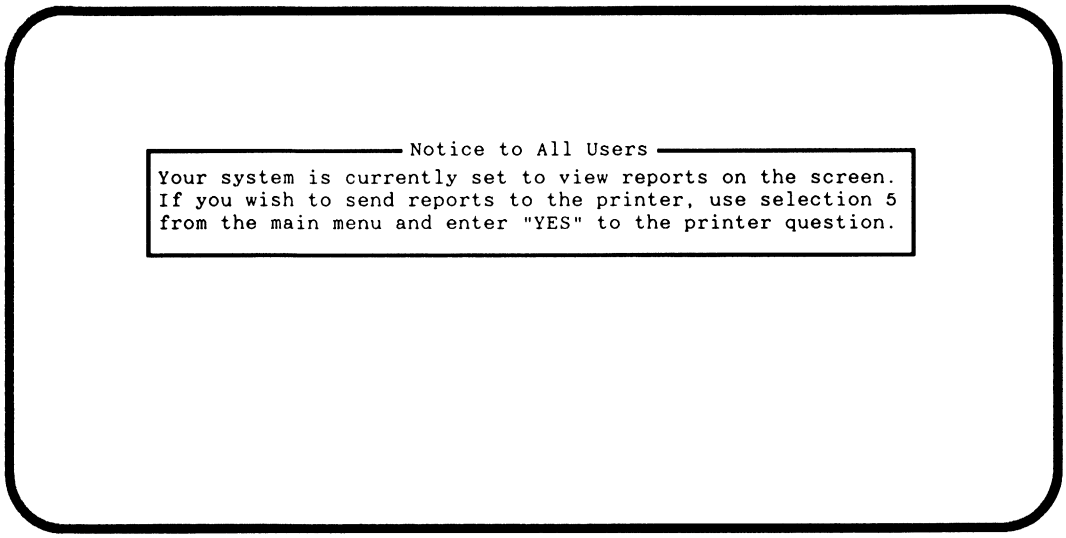


Figure 4–7: A Sample Text Screen

You can preview the appearance of your form by using the COMMAND→VIEW option, explained later in this chapter.

4.6 DEFINING FORM FILES

The DEFINE→FILE command makes files from your database available to the current form. You must define files before fields from the files can be inserted. Figure 4–8 shows the window that FAST TRACK displays when you select the the DEFINE→FILE command.

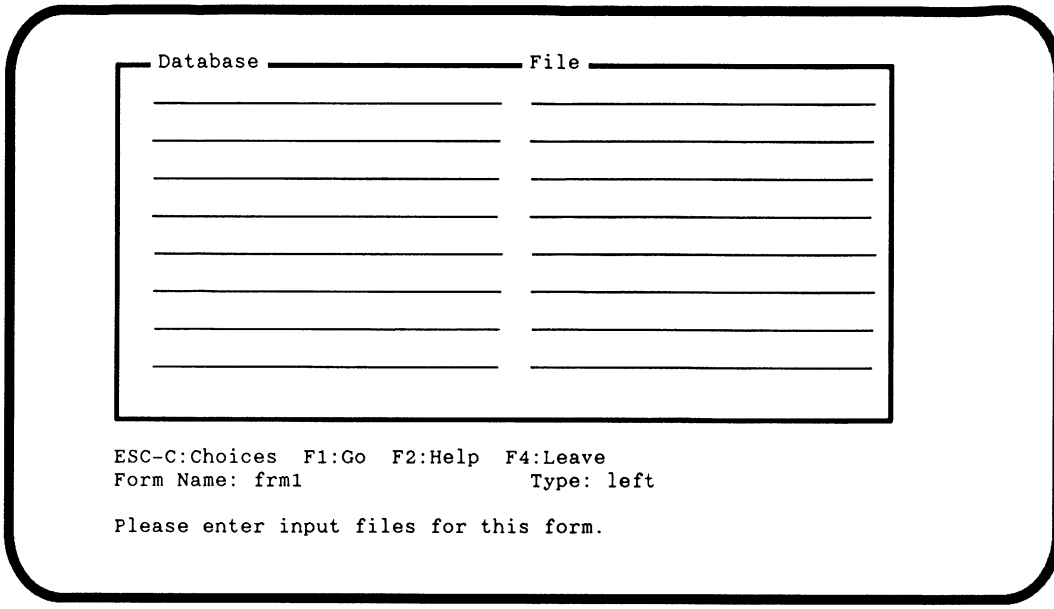


Figure 4–8: The Input Files Window

You can enter up to 12 file names in this window. You type the names of the databases and files you want in this window. If you cannot remember the names of the files for which you want to create your form, press [CHOICES] ([ESC] [C]) to select from the list of available files.

NOTE: Once you have defined files and inserted fields, you use DEFINE→QBF to create a QBF for the form. If you want to use the form for a QBF definition, you can only define one file for it unless the files are one-to-one related. For example, if you start with the `customer` file, you can add `state` or `salesrep` because there is only one each per customer. You cannot add `order`, because there may be more than one order per customer. The reason for this is that it is based on the index. To join, you need a unique index for a common field in the *second* file. For additional information on defining QBFs, refer to Chapter 6 of this manual.

4.7 DEFINING VARIABLE FIELDS

When you use an INSERT→FIELD, INSERT→LABEL, or INSERT→DATA command, the Screen Painter displays a list of the files and fields associated with the form.

One of the choices in the file list is <variable>, as shown in Figure 4-9. With a variable, you can create dynamic fields. These are fields that do not exist in the Data Dictionary definition of the database, but nevertheless can be input by the end-user. For example, you can define a variable field for today's date, then have the application supply a value of `today` for the variable field.

NOTE: If you generate a QBF procedure for the form, you cannot define a variable to appear on the form. If you attempt to define a variable for a form associated with a QBF, FAST TRACK displays an error message.

When you select <variable>, the Variable Definition window in the following figure appears. You enter information about the variable in this window.

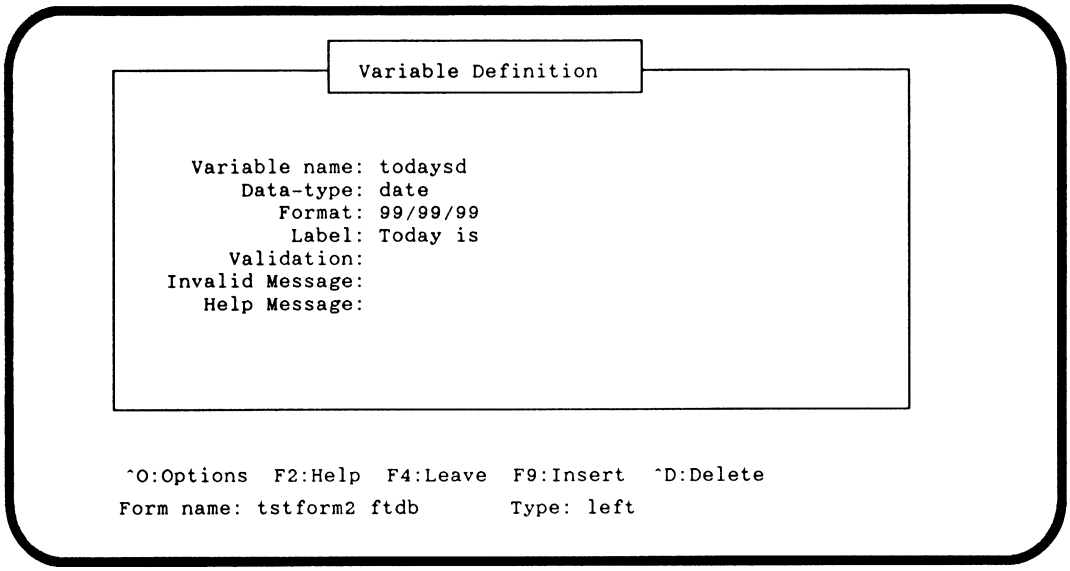


Figure 4–9: Defining a Variable Field for Input

The variable defined in this window is todaysd, which FAST TRACK accesses in order to display the date in your form. To define a variable, enter the following information into the Variable Definition window:

Variable name. Every variable must have a name so you can refer to it in your application. The name can be up to 12 characters long. It can contain any alphabetic, numeric, or special characters, but it cannot contain spaces, nor be a PROGRESS keyword. (You can find the complete list of PROGRESS keywords at the end of the *PROGRESS Language Reference* manual.)

Data-type. The kind of data that can appears in this field. The data type can be one of the following:

- **character** – Alphanumeric data such as character strings. Any printable ASCII character is valid.
- **date** – Any valid date in the time period from 1/1/32768 BC to 12/31/3276 AD.
- **decimal** – A positive or negative number with up to 50 digits and 10 decimal places.

- Defining include files that can perform complex qualifications or lengthy calculations.
- Defining report settings to determine certain default settings, such as: report width, page size, device to receive output of the report, directory to contain report procedures, as well as who can run the report.
- Paginating reports based on their structure.
- Inserting and manipulating report information, such as: field, text, variables, and aggregates.
- Viewing report output.
- Generating report procedures.
- Testing report procedures.
- Saving reports as database objects.
- Leaving the Report Writer.

5.1 DEVELOPMENT PATH

The Report Writer has the most complex development path of all the FAST TRACK modules. To create a report, there are several basic steps, each of which can contain a variety of sub-steps that depend on the type of report you are developing. The following list provides a typical sequence of steps for creating a report:

1. Determine the type of report that you want to create.
2. Select the `Report Writer` option from the FAST TRACK Main Menu and enter a report name and descriptive data in the Report Writer initialization window.
3. Define the structure and other aspects of your report. For example, during this phase of report development, you would typically perform all or some of the following actions:
    - Insert report sections and define files used in the various sections of the report.
    - Use a copy of an existing report in the current database as a template for a new report.
    - Define include files associated with the report.
    - Define report settings, such as the report output characteristics.
    - Define variables.

- Define sort criteria for data in certain report sections.
- Define break groups to logically group data in report sections.
- Paginate the report.

4. Insert and manipulate information on your report. For example, during this phase of report development, you would typically perform all or some of the following actions:
   - Insert and manipulate fields from the database.
   - Insert and manipulate text.
   - Insert and manipulate variables.
   - Insert and manipulate aggregates.
5. View your report using the COMMAND→VIEW command.
6. Use the COMMAND→GENERATE to create a PROGRESS procedure for a report.
7. Test your report procedure using the OTHER→GOTO command.
8. Save your report as an object in the current database using the COMMAND→SAVE command.
9. Save your report in the current database and exit from the Report Writer using the LEAVE→SAVE command.

You do not have to follow the development path to create your reports. It is merely a suggested approach to report development. In fact, you can move back and forth between the definition and insert phases several times as you create your reports.

To complete many of the steps outlined above, you must use the commands on the Report Writer's horizontal command menu. Note that you can only invoke this menu from the Report Writer editing screen. Table 5–1 lists the horizontal menu commands.

Table 5-1: Report Writer Commands

Option	Associated Submenu Options
COMMAND	Copy List Generate View Save
DEFINE	Files Variables Breaks Sorts Includes
INSERT	Field Label Data Value Section Row NextRow Column
REMOVE	Field Section Row Column Picked
MOVE	Next Previous First End Top Bottom Up Down Left Right
PICK	Object Field Area Move Undo
SETTINGS	Report Section Field Mode Page-eject
HELP	*Access help information for the various commands*
LEAVE	Save Quit
OTHER	Opsys Dictionary Main-Menu Reports Goto

Refer to Appendix C for a complete description of all Report Writer commands.

5.2 REPORTS

The FAST TRACK Report Writer lets you use one or several database files in a report and choose the fields that appear in the report. The Report Writer supplies a wide range of commands to format a report any way you like. The following sections introduce several types of reports that you can create with the Report Writer.

5.2.1 Single-File Reports

A single-file report uses the data fields from only one file in the current database. Usually, a single-file report has only the default data section called *Main*. The single database file is defined for use within this section. Once you define a file for the `Main` section, you can insert fields from the file and format the data any way you like. You can sort your data, define break groups for the data section, and perform other formatting operations on your report data.

A single-file report can sometimes have several data sections. This happens only in rare cases. In order for a single-file report to have more than one data section, the same file must be defined for every data section in the report. To define a single file for several data sections, you have to create additional record buffers for each additional data section. Each additional record buffer allows you to access a different record from the same database file. This capability is useful for performing complex logical grouping of records within a single database file.

5.2.2 Multiple-File Reports

A multiple-file report uses the data fields from more than one file in the current database. The introduction of multiple files in one report increases the importance of the report structure. Without some sort of structure, a multiple-file report would be just a jumble of data.

The Report Writer supplies a wide range of tools that allow you to relate files in a report and structure your report to reflect those relationships. There are two basic file relationships that occur. One-to-one relationships, where one record in a file corresponds to only one record in another file. And one-to-many relationships, where a single record in one file corresponds to several records in another file. The use of report sections helps you display these data relationships clearly on your reports.

Report sections are essentially reports within a report. Although you do not have to create new report sections when you have multiple files in a report, the creation of new report sections is an effective way to clearly display data relationships within the report. The creation of a new report section is most effective when used to display a one-to-many relationship between two database files in a report. In this case, the new report section is nested within another report section. Reports that contain nested sections are called *hierarchical* reports.

You can, however, use break groups within a single report section to display a one-to-many relationship between files. A break group consists of a set of records having a common value in a given database field. Every time the value changes, you can display the associated data. For example, if you define the `customer` and `order-file` in one section, and break by `cust-num`, you can display customer data whenever there is a new customer, and display subtotals at the end of the customer data.

5.2.3 Multiple-File Reports with a Single Section

To easily display a one-to-one file relationship on a multiple-file report, define both of the database files within one report section.

For example, suppose that you want to display a corresponding customer name for each invoice record in your database. The customer information is located in a `customer` file, and the invoice information is located in an `invoice` file. Although the `invoice` file contains a customer number, the full name of the customer appears only in the `customer` file. The relationship between the `invoice` file and the `customer` file is a one-to-one relationship. For each customer number in the `invoice` file, there is a corresponding customer name in the `customer` file, as in the following example:

invoice1 customer1
invoice2 customer2
invoice3 customer3
invoice4 customer2
.
.
.

Figure 5-1: Example Report Structure (one-to-one relationship)

With both the **invoice** and the **customer** file defined and related within the same data section, you can insert fields from both of these files in one report data line. This is the clearest way to display a one-to-one relationship between two files.

Within a single data section, you can also display a one-to-many relationship between files. To do this, you define and relate both files within the single data section. Then you have to use break groups to logically group the files according to the established relationship.

5.2.4 Multiple-File Reports with Several Sections

The creation of a new report section is most effective when used to display a one-to-many relationship between two files in a report. In this case, the new report section is nested within another report section. Reports with nested data sections are called hierarchical reports.

Suppose you want to display order information for each customer in a `customer` file. The order information is in an `order` file. The two files are related by an index on the customer number field within both files. There can be several orders per customer. To clearly display a one-to-many relationship between the `customer` file and the `order` file, the report should look something like the following figure:

customer1
  order1
  order2
customer2
  order3
customer3
  order4
  order5
  order6
etc...

Figure 5-2: Example Report Structure (one-to-many relationship)

To obtain the report structure shown in the previous example, you must create a new nested report section. The outer report section, in this case the `Main` section, contains the information from the `customer` file. The nested section contains the information from the `order` file and relates orders to customers.

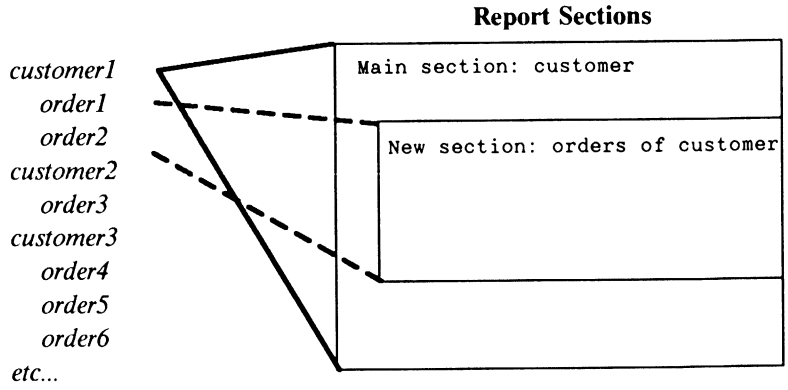


Figure 5–3: Example Hierarchical Report Sections

The previous example shows a typical use of nested data sections. When you run the report, the Report Writer retrieves the first customer record from the database and places it in the `Main` data section. Then, the Report Writer retrieves all order records that are related to the current customer and places them in the `New` data section. After all of the orders corresponding to the current customer are retrieved, the Report Writer repeats the same process for all of the rest of the customer records in the `customer` file.

With the use of nested data sections, comes an important rule of file usage: a nested report section can contain fields from files defined for the parent section. The parent section, however, cannot access fields from files defined in a nested section. In the example, you can only insert fields from the `customer` file into the `Main` data section of the data report. On the other hand, you can insert fields from the `customer` and `order` files within the **New** data section.

Report sections do not always have to be nested within one another in multiple-file reports. You can have sequential sections. For example:

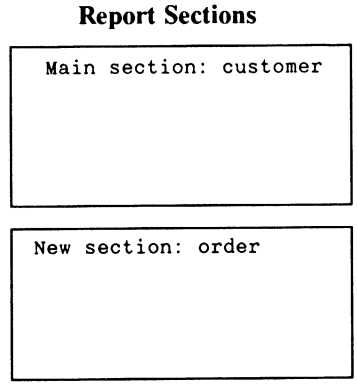


Figure 5–4: Example Sequential Report Sections

If you organize a report as in the previous example, there is no direct relationship between the `customer` file in the `Main` section and the `order` file in the `New` section in the report display. When the Report Writer runs this report, all of the information in the `customer` file appears in the `Main` section and all of the information in the `order` file appears in the `New` section.

You can use any combination of sequential and hierarchical sections to create complex multiple-file reports.

5.3 STARTING THE REPORT WRITER

When you select the `Report Writer` option from the FAST TRACK Main Menu, the Report Writer initialization window appears as follows:

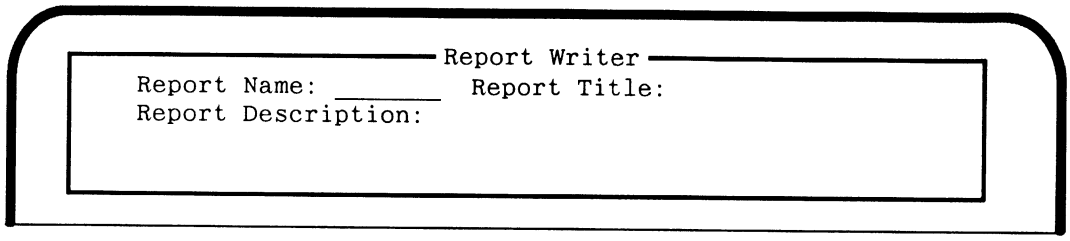


Figure 5–5: The Report Writer Name Window

The Report Writer prompts for the information contained in the following sections:

`Report Name`. The name of your report. The Report Name must conform to the following restrictions:

- It can be no more than seven characters long.
- The name can contain letters, numbers, and the underscore character.

To edit an existing report, you enter its name in the `Report Name` field. If you cannot remember the names of your reports, you can use CHOICES (ESC C) while the cursor is in the `Report Name` field to see available choices.

`Report Title`. The title is for optional documentation. The title appears in any lists that allow you to choose reports. It also appears as a comment in the PROGRESS code when you generate a report procedure.

`Report Description`. This is any optional descriptive information you want to provide about the report. It appears as a comment in the PROGRESS code when you generate a report procedure.

NOTE: Changing a Data Dictionary field definition does not change the definition of that field in any existing Report Writer reports or report procedure (`.p` and `.i`) files.

After you supply the information described, press GO (F1) to begin creating a report. To return to the FAST TRACK Main Menu, press the END (F4) key.

5.3.1 The Default Report Writer Screen

After you name your report in the Report Writer initialization window and press the GO (F1) key, the Report Writer editing screen appears as follows:

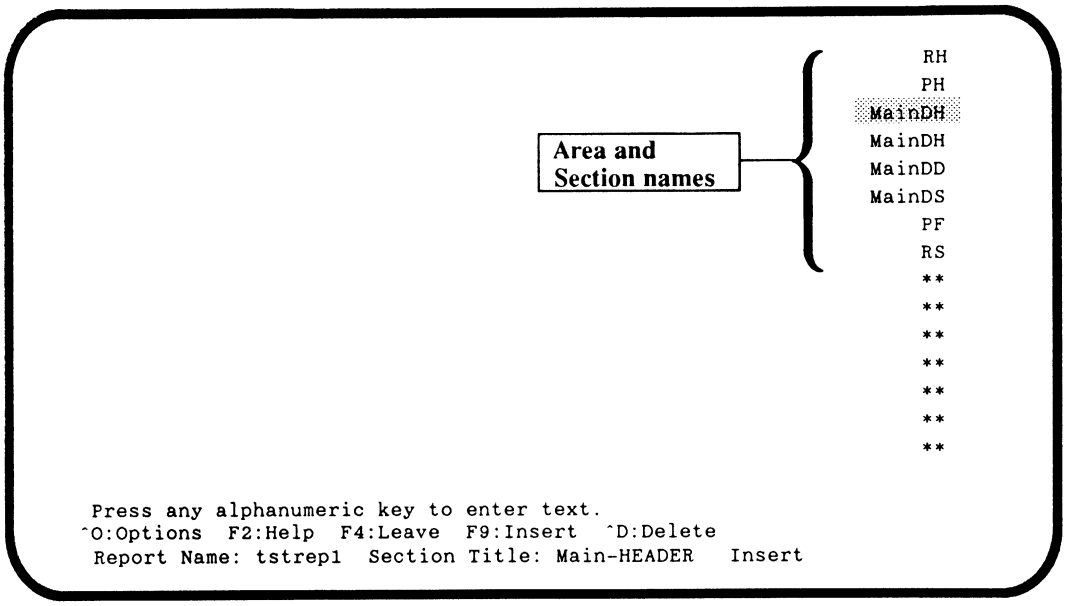


Figure 5-6: The Report Writer Editing Screen

This is the screen FAST TRACK displays every time you create a new report. You see report information at the bottom of the screen, and the abbreviations of area and section names at the right side of the screen.

Familiarize yourself with the information in this window. This is the most important window in the Report Writer. You create all your reports in the Report Writer editing screen.

The commands in the report Writer horizontal menu are discussed throughout the remainder of this chapter. Refer to Appendix C for a complete description of all Report Writer commands.

At the right side of the Report Writer editing screen, the Report Writer displays a column of report section and area names. Every report has designated areas and sections.

A report can have many data sections, but every data section has three areas, each of which begin with the letter **D** for (Detail): the DH, DD, and the DS areas. The DH (Detail Header) area contains the labels for all fields inserted into the corresponding DD area. The DD (Detail Data) areas contain the actual data for your reports. The default detail data area is the MainDD area. Finally, every DD area has a corresponding DS (Detail Summary) area which is used to summarize data in the DD area.

By default, every report has the following areas and sections:

Report Header (RH). Information you type in the RH row appears once at the top of the first page of a report. The report header typically contains a report title, the authors name, the report date, and the report time. You can also enter fields, text, values, and variables into this section.

Page Header (PH). Information that you type in the PH appears at the top of each page in your report. A page header typically contains the page number and report title. You can also enter fields, text, values, and variables into this section.

Main Detail Header (MainDH). These rows appear at the top of a column of data and typically hold all the labels for fields inserted into the MainDD area. When you insert a field into the MainDD area of a report, the label for that field appears in the MainDH for the section. Every new section that you insert into the report has two DH rows. You can also enter fields, text, values and variables into this section.

Main Detail Data (MainDD). Every report includes the Main section. The Main section is the default data section in a report. As with other data sections that you create, the MainDD comprises three data areas, each of which begin with the letter *D* for Detail. These data areas are: DH, DD, and DS.

Main Detail Summary (MainDS). The MainDS row appears at the end of the MainDD area in your report. The MainDS area typically holds information that summarizes the data in the MainDD area. Every new section that you insert into the report will have a DS row. You can also enter fields, text, values, variables, and aggregates into this area.

Page Footer (PF). Information that you type in the PF row appears at the bottom of each page in your report. A page footer typically contains the page number and a report title. You can also enter fields, text, and values into this section.

Report Summary (RS). The RS (report summary) section appears once at the bottom of the last page of a report. It contains aggregates and other information summarizing the data from all the data sections in your report.

Report areas and sections correspond to the way information is organized in the report. Although each of these sections is initially defined to have only one or two rows, you can expand each section and add new sections.

To move between report sections, use the ↓ and ↑ keys. You must enter information into each area that you want to use in a report. The current report section is highlighted. You can only enter information into the current report section. To insert or delete rows in a section, use the appropriate options on the INSERT and REMOVE submenus.

5.3.2 Altering the Default Information Display

There are two commands in the Report Writer horizontal menu that allow you to alter the default information displayed in the Report Writer editing screen. These commands are:

```
SETTINGS→MODE→VERBOSE/BRIEF

SETTINGS→MODE→CURSOR
```

The SETTINGS→MODE→VERBOSE/BRIEF command allows you to turn off the column and area labels at the side of the Report Writer editing screen. You can still view the current section title in the middle of the status line at the bottom of your screen. To display the bar of section and area markers, simply select this command again.

The SETTINGS→MODE→CURSOR command displays the row and column number of the current cursor location. The row and column numbers appear in the menu line near the bottom of the screen. To turn off the row/column display, select this command again.

5.4 DEFINING THE REPORT

During this phase of the report development path, you perform actions that define the report structure and set up preconditions that need to be in place before inserting fields, data, labels, variables, and text into a report. For example, during this phase of report development, you would typically perform all or some of the following actions:

- Insert report sections and define files used in the various sections of the report.
- Use a copy of an existing report in the current database as a template for a new report.
- Define include files associated with the report.
- Define report settings, such as the report output characteristics.
- Define variables.
- Define sort criteria for data in certain report sections.
- Define break groups to logically group data in report sections.
- Paginate the report.

The definition stage is the most important step in the development path. Some of these actions must take place before inserting data into a report. For example, you cannot insert fields from a database file into a report until you define the file for use in the report. The following sections introduce a number of commands to define the report structure and other aspects of your report. You access these commands from the horizontal command menu.

5.4.1 Defining Files

The DEFINE→FILES command makes database files available to a report. Files are most commonly defined for the data sections in reports. You can define a maximum of ten files per section. Before selecting this command, position the cursor within the section for which you are defining a file. When you select the DEFINE→FILES command in a report section, the Input Files window appears as shown in Figure 5-7.

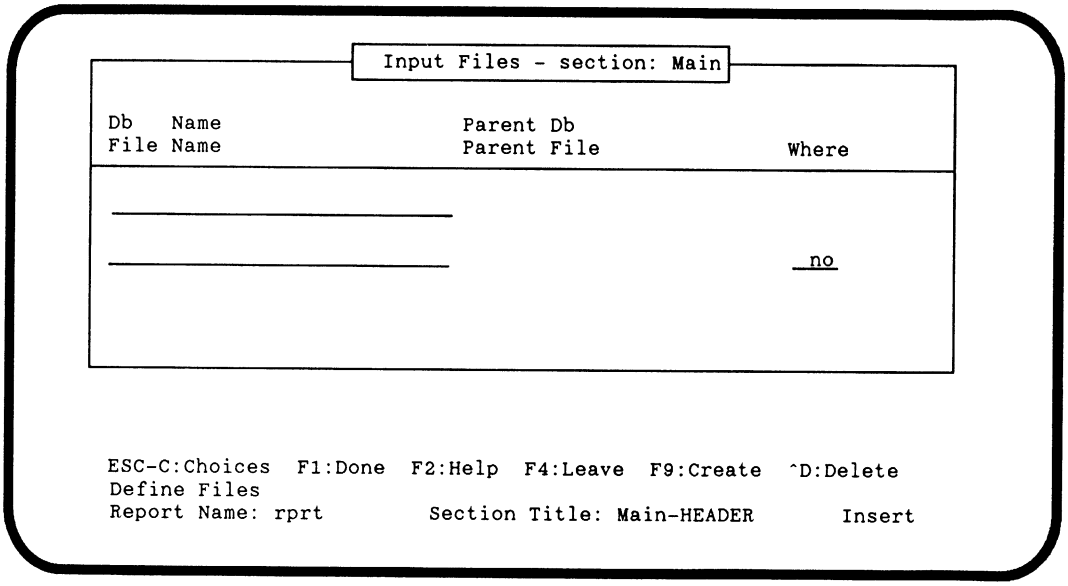


Figure 5-7: The Input Files Window

To move between the fields on this window, use the TAB key. Use the ↓ key to create a new file input line. When you have finished defining files for use in the current report section, press the GO (F1) key to accept your entries. The Input Files window contains the following fields:

Db Name. Enter the name of the database in this field. If you use CHOICES (ESC C) to select the File Name, the Db Name will be automatically entered.

File Name. Enter the name of a database file in this field. The filename must be contained in a connected database. If you cannot remember the names of files available in the currently connected databases, press CHOICES (ESC C) to see a list of available files and select one.

Parent File. As stated before, there can be one-to-one and one-to-many relationships between files in your database. When you define a database file for use in a report, you can also use these relationships to effectively display your data. The `Parent File` field allows you to use file relationships that are already established in your database.

A *Parent File* is any file previously defined in the current report related to the current input file by a unique index in the database schema. In other words, all index fields in a unique index in one of the files must exist in the other file. They must also have the same name and data type. If there is more than one unique index relating the parent file to the current input file, you have to explicitly join files by using the `Where` option described in the next section.

Parent Db. This file indicates the database where the `parent file` is defined. This field allows you to define the relationship across the databases.

To display a list of files that are currently in the report and related to the current input file by an index in the database schema, use the CHOICES (ESC C) key. It is important to note that you cannot specify a parent file for the first file defined in the outer most section of a report.

Where. When defining a file for use in a report, it is often useful to qualify the data you want to appear in the report. Use the `Where` field to accomplish either one or both of the following tasks:

- Restrict the data displayed in the report to a subset of the records in the current input file.
- Create a relationship between the current input file and another previously defined file in the report.

To accomplish these tasks, FAST TRACK allows you to enter a qualification that determines the data displayed in the report from the current input file. There are two ways to specify a qualification:

- Enter a structured qualification.
- Enter a free form qualification.

By default, the `Where` field has a **no** value. This means there is no qualification on the data from the current input file.

To enter a qualification for the current input file, press RETURN or TAB after entering a file name or parent file. FAST TRACK displays the Qualification window, as shown in the following figure:

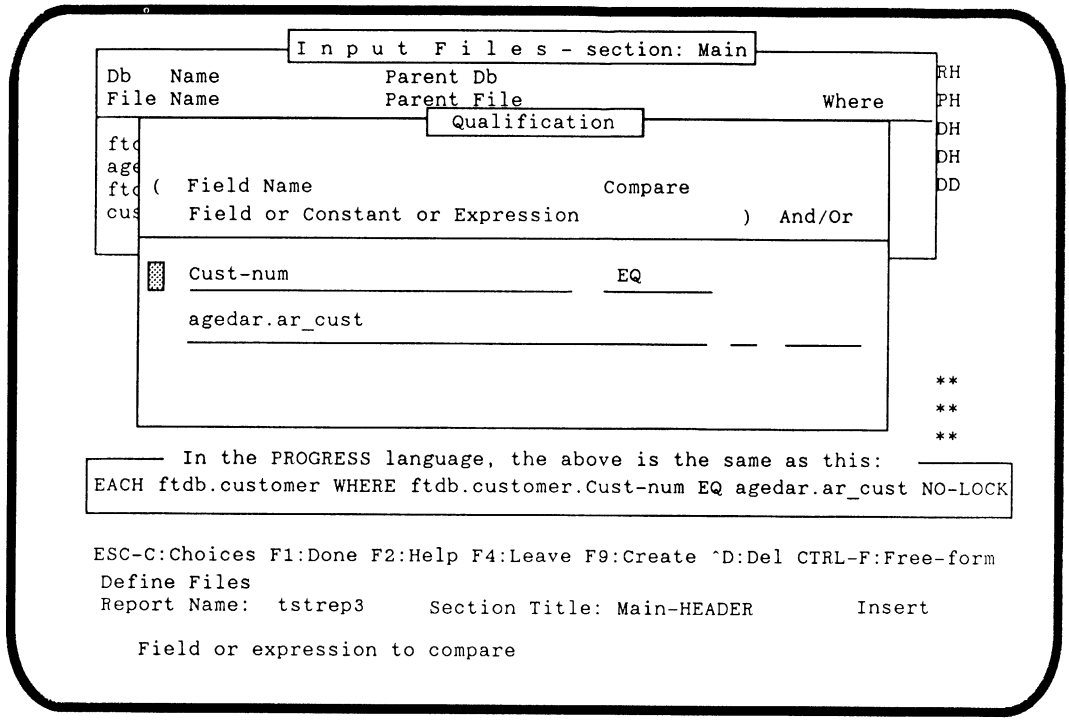


Figure 5–8: Qualification Window

5.4.2 Qualifications

A qualification can contain up to 50 comparison expressions. To create a new line for a comparison expression in the Qualification window, press the ↓ key. Every comparison expression has the following three parts:

- The name of the field from the current input file that you want to test.
- A comparison operator that determines the kind of test to perform.
- The name of another field, a constant, or an expression with which the first field is compared.

The top line of the Qualification window shows the format of a qualification. The Qualification window has six fields:

(— Enter up to two parentheses to group expressions. For every left parenthesis, there must be a right parenthesis. You can group expressions over several input lines in the Qualifications window.

Field Name — Enter the name of the field upon which you want to base your qualification.

Compare — Enter a comparison operator in this field.

Field or Constant or Expression — Enter the name of a field, constant, or expression to compare.

NOTE: When you enter a constant date, you should always enter it in the order month, day, then year, even if the display format is different. Also, constant logicals should always be entered as yes/no or true/false, not the format values. For example, if the format of field shipped is shipped/not shipped, you wouldn't want to check for shipped = "shipped", but you would want to check for shipped = true.

And/Or — A logical connective that relates comparison expressions in a qualification.

) — Enter up to two parentheses to group expressions. For every right parenthesis, there must be a left parenthesis. You can group expressions over several input lines in the Qualification window.

Here's an example qualification that relates a current input file to a previously defined file in a report:

```
Cust-num EQ agedar.ar_cust
```

This example relates the current input file, `customer`, to a predefined file, `agedar`. Both of these files are from the FAST TRACK demonstration database used in the *PROGRESS FAST TRACK Tutorial.* These files are not related in the demonstration database schema, yet both customer number fields can be used to establish a file relationship through the `Where` field.

The previous example illuminates this relationship. The example relates the `Cust-num` field of the `customer` file with the `ar_cust` field of the `agedar` file. Both of these files contain the customer number data; however, they have different names.

A typical qualification that limits data displayed from the current input file on your report would look like this:

```
st EQ "MA"
```

This qualification specifies that only records from the current input file in which the state field (`st`) equals "MA" is shown in the report. In this case, the `Where` is not used to create a relationship between files in the report and you can use the `Parent File` field to establish a relationship.

As you enter your comparison expression into the Qualification window, FAST TRACK translates it into a PROGRESS FOR EACH statement. This PROGRESS statement appears at the bottom of the screen. (For more information about the PROGRESS FOR EACH statement, see the *PROGRESS Language Reference* manual.)

You can press the CHOICES (ESC C) key to display a list of acceptable values to enter into the current field and select one. For example, if you are in the Compare field and you press the CHOICES (ESC C) key, the following list of comparison operators appears:

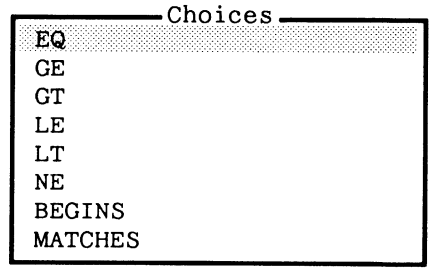


Figure 5–9: The Compare Field Choices

After you make an entry in Field Name field and press RETURN, the Report Writer automatically fills in the Compare field with a default comparison operator for the particular data type of the field. The defaults are:

- EQ for integer, decimal, and data data types.
- BEGINS for the character data type.

When you have finished entering a qualification for the current input file, press the GO (F1) key to enter the qualification and return to Input Files window.

5.4.3 Comparison Operators

This section explains the available comparison operators. Note that the comparison operators compare *like* data types. (For more information about PROGRESS data types see the *PROGRESS Language Tutorial.*)

All comparison operators work with all data types except for the operators BEGINS and MATCHES, which only work with the character data type. The following sections describe the comparison operators:

EQ or = The EQ or = operator tests for *equality* between two values. The following are typical examples:

```
cust-num = 23
name EQ "Hoopla Basketball"
sales-region = "West"
```

The first example specifies the record for customer number 23; the second example specifies the record for the Hoopla Basketball company; the third example specifies the records of all customers in the western sales region.

GE or > = The GE or >= operator tests whether one value is *greater than or equal* to another. For example

```
mnth-sales[7] GE 10000
```

This comparison specifies all the records of customers who had sales equal to or surpassing $10,000 for the month of July.

GT or > The GT or > operator tests whether one value is *greater than* another. For example:

```
max-cred GT mnth-sales[3]
```

This comparison specifies the records of all customers whose sales for the month of March exceeded their maximum credit.

LE or < = The LE or <= operator tests whether one value is *less than or equal* to another. For example:

```
curr-bal <= max-cred
```

This example specifies the records of all customers whose current balance is less than or equal to their maximum credit.

LT or < The LT or < operator tests whether one value is *less than* another. For example:

```
ytd-sls LT 30000
```

This comparison specifies all the records of customers whose year-to-date sales are less than $30,000.

NE or < > The NE or <> operator tests whether two values are *not equal*. For example:

```
discount <> 0
```

This comparison specifies the records of all customers whose discount is *not equal* to 0.

BEGINS The BEGINS operator specifies all records in a file that contain a character field beginning with a specified character string.

```
city BEGINS "o"
```

This qualification finds the records for all customers from cities beginning with "o."

MATCHES The MATCHES operator finds all records in a file that contain a character field matching a specified character string. This operator accepts

wildcard characters: a period (.) in the specified character string matches *any* single character, and an asterisk (*) matches any string of characters. Thus, *"p.t"* matches both *"pit"* and *"pot"* and *"p*t"* matches both *"peat"* and *"pout"* as well as *"pit"* and *"pot"*. The following example illustrates the use of the MATCHES operator:

```
city MATCHES ".uf*"
```

This qualification finds the records of all customers who come from cities that begin with any single character followed by "uf," followed by any string of characters. The city of Buffalo is a valid example.

5.4.4 Logical Operators

More complex qualifications can be constructed in any of several ways. One way is by joining several simple comparison expressions with And, And Not, Or, or Or Not. You enter each expression in a connected qualification on a separate line in the Qualifications window, as shown in the following figure:

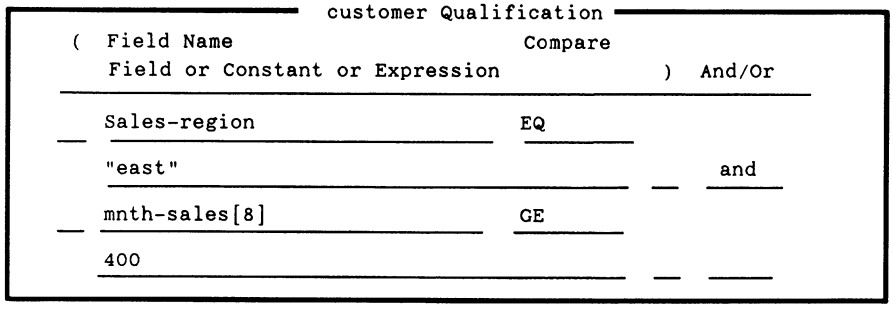


Figure 5-10: Example Qualification

In the Qualification window, enter either **And** or **Or** in the field titled And/Or. Note that you enter the **Not** operator in the And/Or field as well. You can press CHOICES (ESC C) for a correct list of choices. The following sections explain the And, Or, and Not operators:

And When comparison expressions are connected by And, *both* expressions must be true for a record to be included in the report. For example:

```
sls-rgn EQ east and mnth-sales[8] GE 400
```

Or When expressions are connected by Or, only one of the expressions must be true in order for the record to be included in the report. For example:

```
city EQ "Boston" or city EQ "San Diego"
```

Not The negation operator is Not. It is always used with the And or Or operators. For example:

```
st EQ "CA" and not city EQ "San Diego"
```

This example finds the records of customers in all California cities other than San Diego.

```
st EQ "CA" or not city EQ "Boston"
```

This qualification finds the records of all customers in California as well as the records of all customers in all other cities outside of California, except Boston.

After entering the **And/Or** operator at the end of the line, press ↓ to move to the next line.

5.4.5 Order of Evaluation

Be careful when formulating qualifications consisting of several joined clauses with negatives. When you begin to combine more than two comparison expressions, you can introduce apparent ambiguities. Consider the following qualification:

```
st EQ "MA" and city EQ "Boston" or city EQ "San Diego"
```

This qualification searches for those customers who live either in Boston, Massachusetts or in San Diego, where San Diego can be in any state. The logical connective **And** always dominates **Or**. To search for those customers who live in Boston, Massachusetts or San Diego, Massachusetts, use the following qualification:

```
st EQ "MA" and (city EQ "Boston" or city EQ "San Diego")
```

The following table shows the order of precedence for the PROGRESS comparison and logical operators:

Table 5–2: Precedence of Operators

Name of Operator	Precedence
GE, GT, LE, EQ BEGINS, MATCHES	4 (highest)
NOT	3
AND	2
OR	1 (lowest)

Logical expressions within parentheses are evaluated before expressions that lie outside the parenthesis. (For additional information on precedence, see the *PROGRESS Language Reference* manual.)

5.4.6 Free Form Qualifications

Free form qualifications allow much more freedom in specifying a qualification than is possible with the structured Qualification window. When the cursor is in the Qualification window, press CTRL-F. The Free Form Qualification window appears as follows:

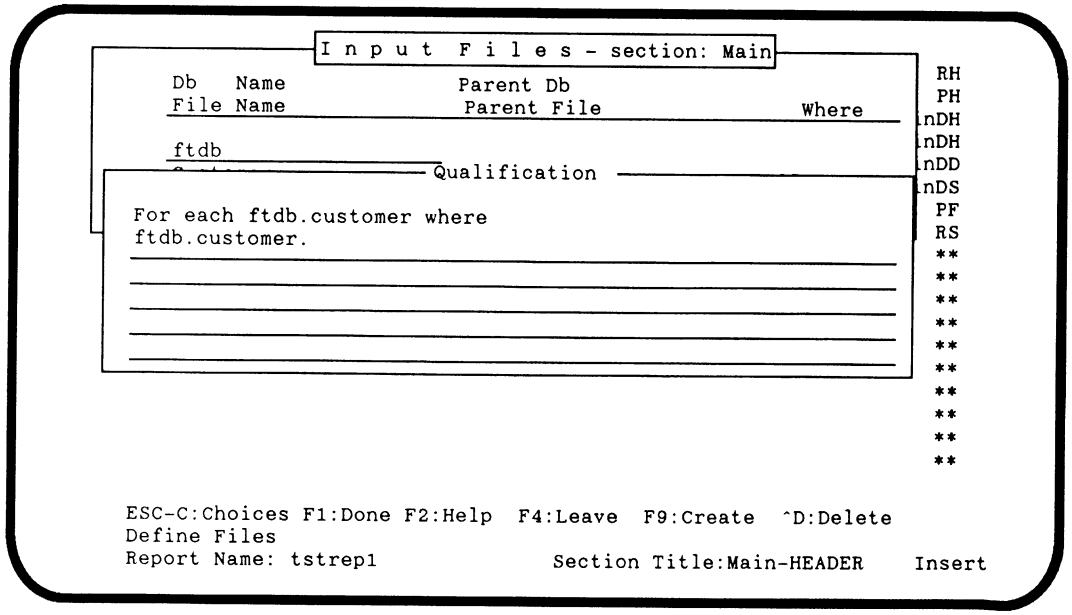


Figure 5–11: The Free Form Qualification Window

Notice that there are no set fields in which to enter information. You enter your qualification in the Free Form Qualification window, simply by typing your qualification.

Because this window is unstructured, you can use it to create qualifications that:

- Require more than two levels of parentheses.
- Require an expression instead of a field name to the left of the comparison operator.
- Require an expression longer than 32 characters to the right of the comparison operator.

Suppose you are looking for those customers with a zip code beginning with the digits 1 and 3. An appropriate qualification for this type of search is difficult to design using the structured Qualification window. Using a free form qualification, however, allows you to take care of this type of search with the following PROGRESS statement:

```
TRUNCATE ( Zip / 1000, 0 ) = 13
```

5.4.7 Record Buffers

By default, PROGRESS opens only one record buffer for each file accessed from the current database. Remember, FAST TRACK is a PROGRESS application. With only one record buffer for a particular database file, you can access only one record at a time from that file. You can change this default by defining an additional buffer for a database file. In this way, you can access two or more different records from the same file at the same time.

When defining a second record buffer for a file in a report, select the DEFINE→FILE command just as you would for defining a file for a particular report section. When the Input Files window appears, use the following syntax in the File Name field to define the second buffer for a file:

SYNTAX

```
buffer-name(database-file)
```

where *buffer-name* is the name of the second buffer for the file *database-file*. The *database-file* entry must be the actual name of a file in your current database. The *buffer-name* cannot be the name of an existing buffer or file. The *database-file* entry does not have to be accessed from another section within the report through the default record buffer for that file. For example, to define another record buffer for the customer file in the FAST TRACK demonstration database, enter the following in the File Name field:

```
custbuf(customer)
```

As with any normal file definition, you can qualify the data that appears in the report from the buffer. This capability is useful for performing complex logical groupings on records within the same database file.

5.5 INSERTING AND DEFINING REPORT SECTIONS

Report sections are essentially reports within a report. All reports created with the Report Writer have one data section by default. This section is called **Main**. Although you do not have to create new report sections to have several files in a report, the creation of new report sections is an effective tool that allows you to clearly display data relationships in your report.

There are several commands in the Report Writer horizontal menu that allow you to insert, view, remove, and manipulate sections in a report. The following sections identify and explain how to use each of these commands.

5.5.1 Inserting Report Sections

The `INSERT→SECTION` command allows you to insert a section in your report. When you select this command, the following window appears:

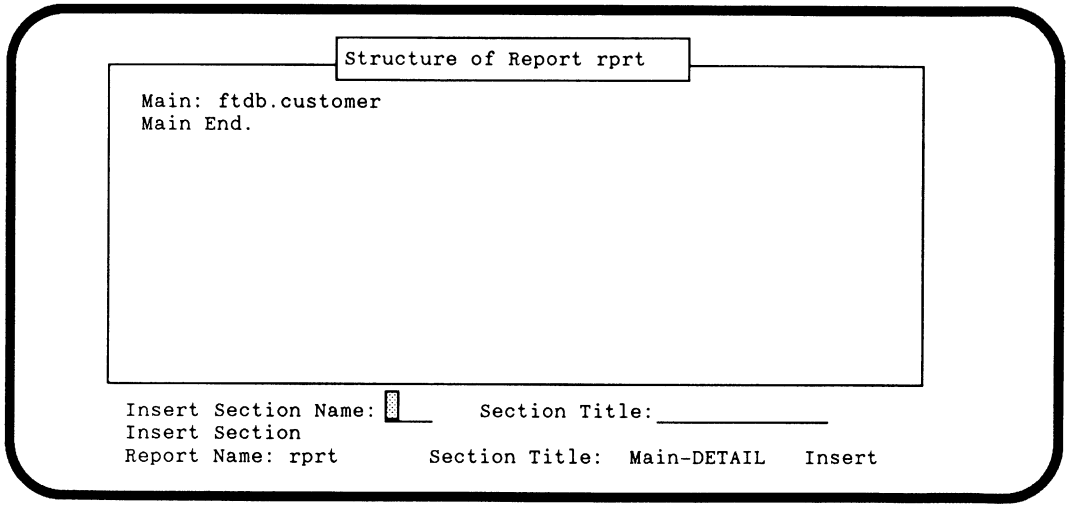


Figure 5-12: Report Structure Window

This figure shows the structure of a report named `rprt`. This report has only the `Main` section. The `Main` section has the `ftdb.customer` file defined in it.

You are prompted for the section that you are inserting. In the `Insert Section Name` field, enter a section name of up to four characters and press RETURN. This entry appears in the margin of report areas when you return to the Report Writer editing screen. In the `Section Title` field, enter a section title of up to 15 characters and press the GO (F1) key. When you highlight this new section in the Report Writer editing screen, the entry in the `Section Title` field will appear on the bottom line of your screen. For example, the section title `Main-Detail` appears on the last line of the preceding figure.

After you press the [GO] (F1) key, the Report Writer automatically nests the new report section within the Main section on the current report.

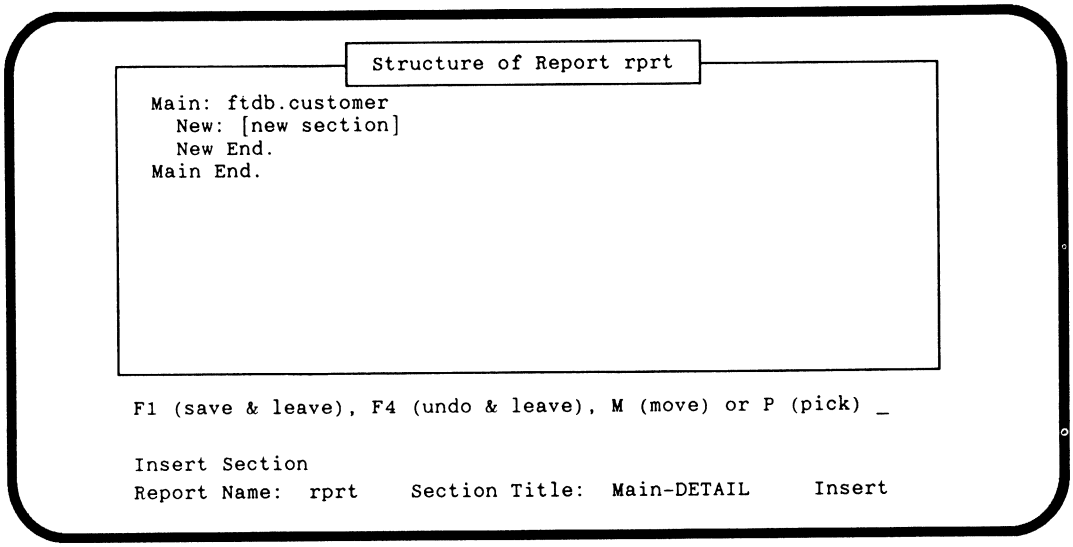


Figure 5-13: Report Structure Window

At this point, the Report Writer lists four Report Structure options:

- Press [GO] (F1) to save the report structure as it currently appears in the Report Structure window and enter the Input Files window to define the files for the newly created section.
- Press the [END] (F4) key to cancel all changes to the report structure and return to the Report Writer editing screen.
- Type **M** and press [RETURN] to move the new section to a different position within the report. The Report Writer automatically highlights the new section. To move the highlighted section, use the [↓] and [↑] key to move the cursor to the line where you want the new report section to begin. Then press the [INSERT] (F9) key.

  If you want to move a section other than the new section, press the [CLEAR] (F8) key to cancel the current selection. Then use the [↓] and [↑] key to highlight another report section and then press the [GO] (F1) key and the Report Writer returns you to the Report Structure options.
- Type **P** and press [RETURN] to pick and move any section on the current report. Use the [↓] and [↑] key to highlight a report section and then press the [GO] (F1) key to pick the section. To move the highlighted section, use the [↓] and [↑] key to move the cursor to the line where you want the new report section to begin.Then press the [INSERT] (F9) key.

If you want to move a section other than the new section, press the [CLEAR] (F8) key to cancel the current selection. Then use the [▼] and [▲] key to highlight another report section and then press the [GO] (F1) key and the Report Writer returns you to the Report Structure options.

If you enter a new report section and save it to the report structure using the [GO] (F1) key, the Input Files window appears automatically. Define files for the new section just as you would using the DEFINE→FILES command. Remember to take into account the file relationships that you want to display on your report. You can define a maximum of ten files per report section. Once you finish defining files for the new section, press the [GO] (F1) key to return to the Report Writer editing screen. The new section appears in the margin of area and section names in the Report Writer editing screen.

5.5.2 Manipulating Report Sections

The SETTINGS→SECTION command allows you to change the default setting for the current report section. Before selecting this command, position the cursor in the section that you want to alter. When you select this command, the Section Settings window appears as follows:

```
                    Section Settings
Section Name      : Main          Title : Main
Label (Top/Side)  : top
Page-eject        : no
```

Figure 5-14: The Section Settings Window

This window allows you to modify the following section settings:

Section Name. Enter a new section name of up to four characters in length. This name appears in the margin containing report area and section names in the report Writer editing screen. The default is the current section name.

Title. Enter a string of up to 15 characters in length. The section title appears at the bottom of the Report Writer editing screen to let you know what section you are working in. The section title is optional. The default is the current section name.

Label. Enter either **top** or **side**. This setting allows you to specify whether you want to insert your data with top labels or side labels in the current section of your report. Once you insert a field into the current section, you cannot change this setting.

Page-eject. Enter either **yes** or **no**. This setting allows you to specify whether you want a page-eject after this section of your report.

After entering the settings you want for the section, press the [GO] (F1) key to save these changes to the report.

5.5.3 Removing Report Sections

The REMOVE→SECTION command allows you to remove any section from a report. When you select this command, the following figure appears:

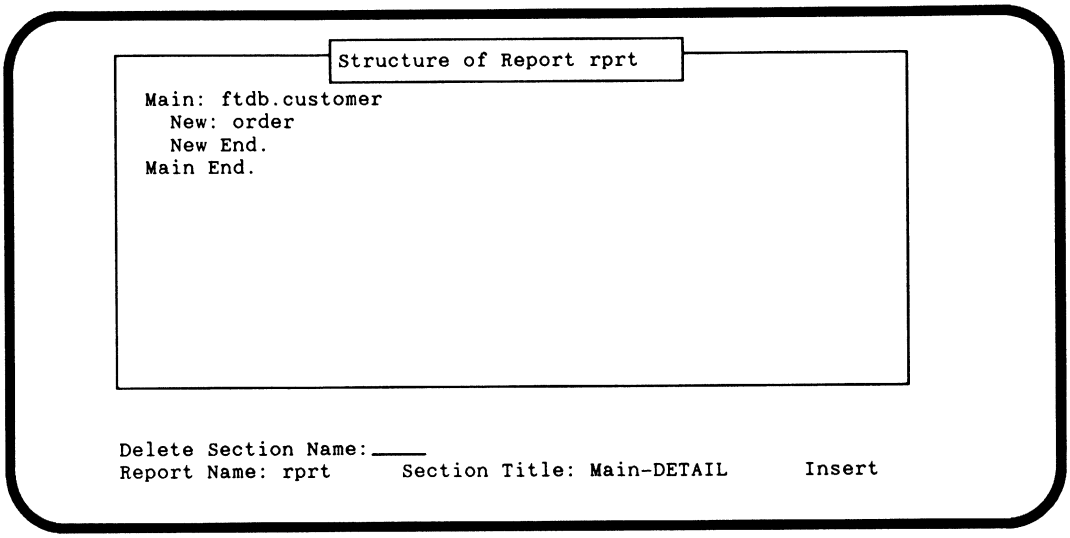


Figure 5–15: Delete Section Window

To delete a section from the current report, enter a section name into the Delete Section Name field and press RETURN. The Report Writer highlights the specified section and displays the following prompt at the bottom of the screen:

```
Are you sure that you want to delete this section ? (Y/N)
```

Type **Y** and press RETURN to delete the highlighted report section and return to the Report Writer editing screen. Type **N** and press RETURN to cancel the deletion and return to the Report Writer editing screen.

5.5.4 Viewing Your Report Structure

The COMMAND→LIST command displays the logical structure of the current report. When you enter this command, the Report Structure window appears as follows:

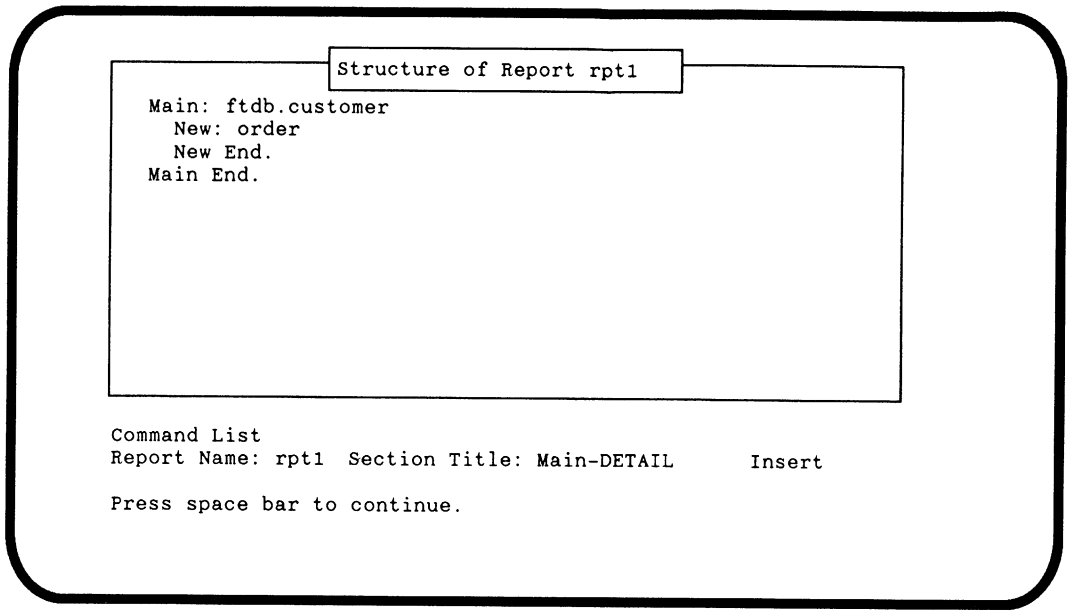


Figure 5-16: Report Structure Window

This window shows the structure of a report named `rpt1`. This report has two sections: `Main` and `New`. The `Main` section has the `customer` file defined in it. The `New` section is nested within the `Main` section; it has the `order` file defined in it. After viewing your report structure, press the [SPACEBAR] to return to the Report Writer editing screen.

5.6 USING REPORTS AS TEMPLATES

The COMMAND→COPY command allows you to use other reports as templates to create new reports. This command copies an existing report object from the current database into the current report. When you select this command, the following window appears:

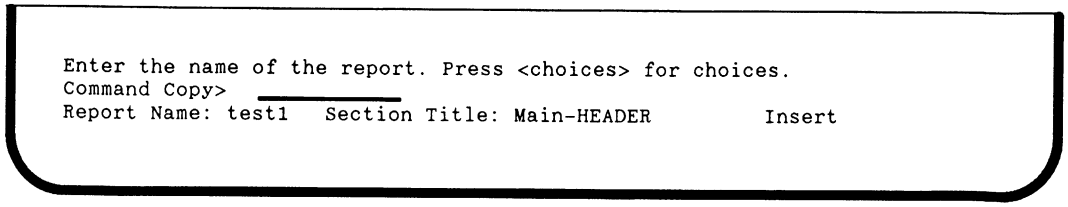


Figure 5-17: Report Copy Window

At `Command Copy>` prompt, enter the name of the report you want to copy. You can also press [CHOICES] ([ESC] [C]) for a list of the reports available for copying. After you select a report, press the [GO] (F1) key. The Report Writer copies the report you specify into the current report, overwriting everything in the current report.

5.7 DEFINING VARIABLES

You can define a variable as a means to temporarily store and display calculated data in a report. You can also define a variable to perform complex breaks and sorts.

The data stored in a variable is not part of your database, but may be based on calculations using data from your database or elsewhere. To make a variable available for use in a report, define it using the DEFINE→VARIABLES command. When you select this command, the Report Writer displays the Define Variables window as shown in Figure 5–18.

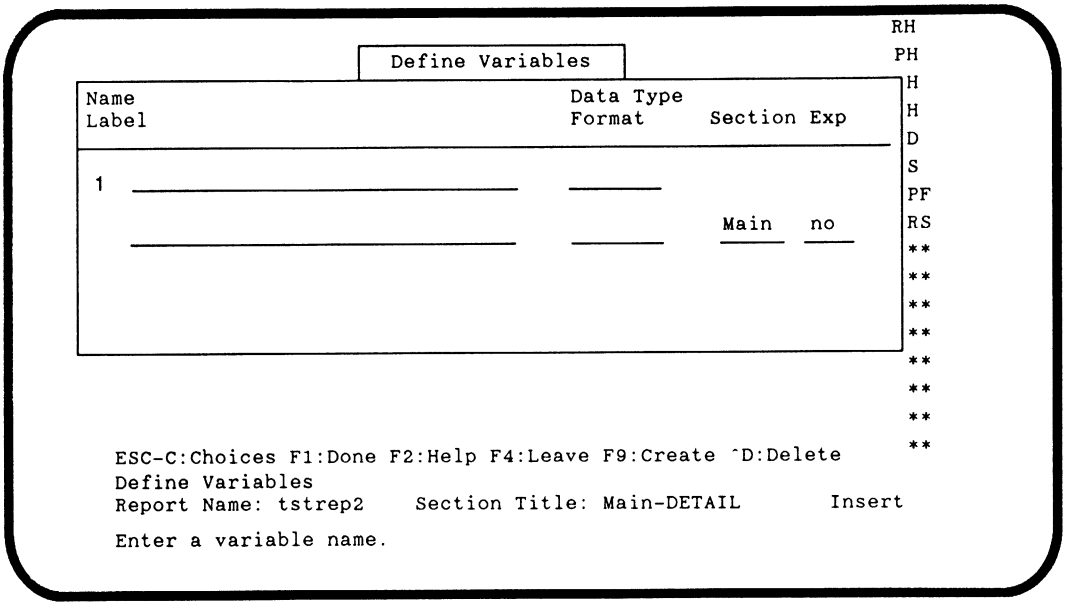


Figure 5–18: Define Variables Window

The Define Variables window has the following fields:

Name. The name of the variable you want to define. It must conform to the following constraints:

- The name can be up to 12 characters long.
- It can contain any alphabetic, numeric, or special characters, except spaces.
- The name cannot be a PROGRESS keyword.

You can find the complete list of PROGRESS keywords at the end of the *PROGRESS Language Reference* manual.

`Data type.` Enter one of the following data types for the variable:

Character Non-numeric data, such as character strings.

Date Any valid date in the time period from 1/1/32768 BC to 12/31/32768 AD.

Decimal A positive or negative number with up to 50 digits and 10 decimal places.

Integer Any positive or negative whole number.

Logical Anything that can have one of two possible values, such as Yes/No or Up/Down.

To specify a data type, enter the first letter or two. FAST TRACK recognizes a unique choice and moves to the next field.

`Label.` Enter a label up to 20 characters in length. This label will appear with the variable when it is inserted into the report. A label is not required.

`Format.` This field contains the default data format for the data type you entered into the `Data type` field. Change the format if necessary. See the *Programming Handbook* for more information about data formats.

`Section.` Enter the section name for which you want to insert the variable. The section name must be in the current report. When you specify a section, the variable is available to insert in that section and all sections nested in it. If you leave the field blank, you can use the variable in all report sections.

`Exp.` When the cursor moves to the `Exp` (expression) field, the Expression window appears on your screen as follows:

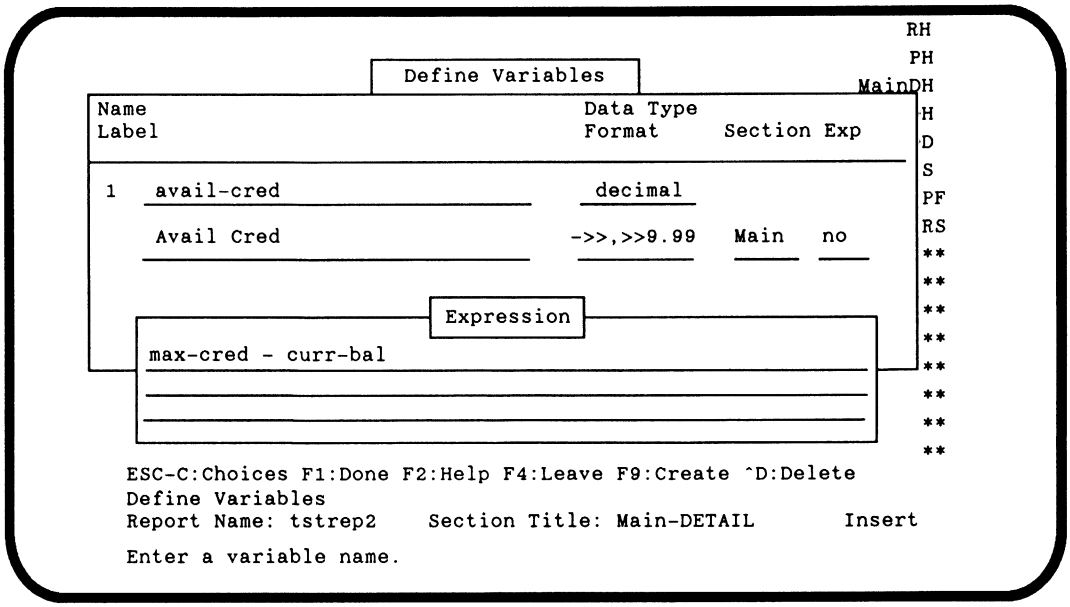


Figure 5–19: Define Variables Window (with Expression Window)

You can specify a wide variety of calculations for the value of a variable. You can use any arithmetic operator, or combination of operators in an expression. You can also combine constants (fixed values that you use in the expression) and database field values.

The Report Writer also lets you use the fields from the database files defined for the sections specified for the variable. See the *Programming Handbook* for a listing of the various operators that you can use and the precedence of those operators.

After you enter the expression in the Expression window, press GO (F1). The entry under Exp in the Define Variables window changes to yes.

You can define several variables in the Define Variables window. To enter additional variables, press ↓ to enter a variable after the current variable definition or press the INSERT (F9) key to enter a variable before the current variable definition.

To delete a variable definition, position the cursor on a variable definition line and press the DELETE (F10). After you delete a variable definition, you can no longer insert the variable into the report. The Report Writer allows you to remove any field definitions that are dependent on the variable from the report display.

Variables are evaluated in the order in which they are inserted into a section. For example, a variable **a** equals **1** and a variable **b** equals **a + 1**. To display these variables in a report section, you must insert variable a before inserting variable b. The order of variable evaluation is indicated in the Define Variables window by the number to the left of each variable definition line.

When you finish defining variables, press [GO] (F1) and the Report Writer editing screen appears. Now you can insert the variables you defined with the INSERT→FIELD command.

5.8 DEFINING BREAK GROUPS

The Report Writer allows you to break report data into logical groups for display. To accomplish this task, you must define break groups. A *break group* is data grouping based upon the value of a specified data field or variable.

In other words, all records in a file that have a common value for the specified field are grouped together. The Report Writer allows you to define break groups within any data section in your report. With a break group, you can group data from one or many database files.

Sometimes you may need to break by an item not defined as a field. For example, you could break by the first letter of a state code. To do this, define a variable with the corresponding expression, and break by the variable.

The DEFINE→BREAKS command allows you to define a break group within the current report section. Before selecting this command, position the cursor in the section where you want the break group to appear. After you select the DEFINE→BREAKS command, FAST TRACK displays the break groups window shown in the following figure:

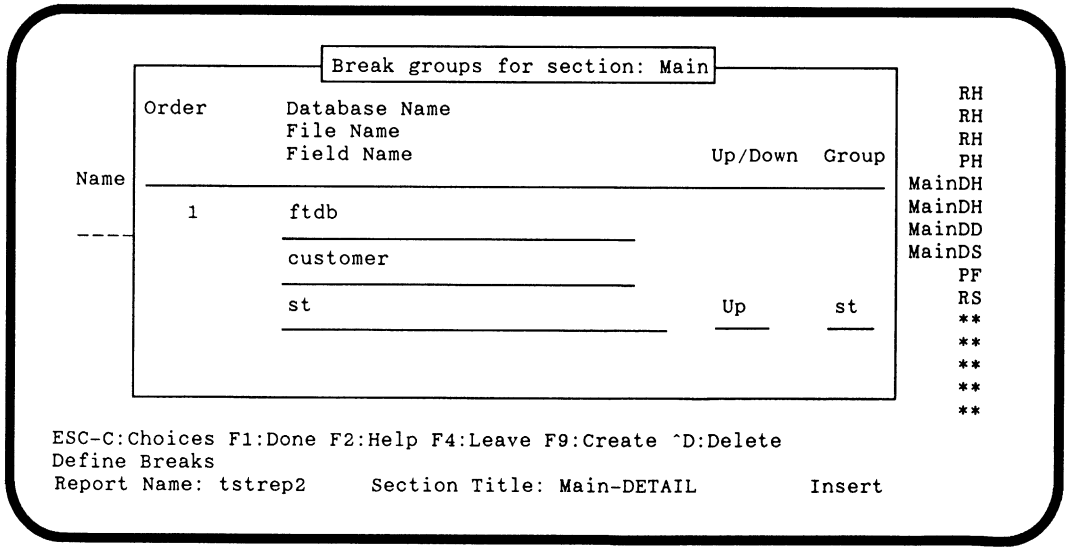


Figure 5-20: Break Groups Window

This window has the following fields:

Order. The Report Writer enters a default order for you. This item is the order in which the Report Writer processes the breaks. You cannot change this number.

`Database Name`. This is the name of the database that contains the file.

`File Name`. This is the name of the file or variable for which you want to define a break group. This file must be defined for the current report section. If you want to break by variable, enter **<variable>**.

`Field Name`. The name of the field or variable by which you want the data grouped. The field must be in the file specified in the `File Name` field. The field specified need not appear in the report.

`Up/Down`. Enter **Up** if you want the break group sorted in ascending order (from a to z or from the lowest to the highest number). Enter **Down** if you want the break groups sorted in descending order (from z to a or from the highest to the lowest number).

`Group Name`. Enter a group name of up to four characters in length. When you return to the Report Writer editing screen, this group name appears in the current report section. A group header and footer also appears.

You can define multiple break groups on a file in a report section. The order in which you enter break group definitions into the break groups window determines the order in which the break groups are evaluated when displaying the report data.

To enter a break group after the current break group definition, press the [↓] or [RETURN]. Use the [INSERT] (F9) key to insert a new break group definition before the current break group definition.

To delete a break group, position the cursor on a break group definition line within the Break-groups window and press [DELETE] (F10) key. If you have inserted fields in the break group, the Report Writer asks you to confirm the deletion of the break group definition. When you finish entering break groups, press [GO] (F1) to save the break group information and return to the Report Writer editing screen.

A group header (GH) section and a group summary(GS) section appear in the list of section and area names on the right side of the Report Writer editing screen. You can use any of the `INSERT` and `REMOVE` submenu options to add or delete new rows to these report sections.

In the break group header (GH) section, use the `INSERT→FIELD` command to insert the field by which you want to break the report. You can insert an aggregate into the break group summary (GS) section to summarize the data in the group. When you run the report, the data is divided into the break groups you defined.

5.9 DEFINING SORTS

The Report Writer lets you to sort information in any report section. You can sort information in ascending or descending order. Alternatively, you can sort data by an item not defined as a field. For example, you could sort by the second letter in a customer code. To do this, define a variable with the corresponding expression, and sort by the variable.

Before selecting the DEFINE→SORTS command, position the cursor in the report section you want to sort. When you select the DEFINE→SORTS command, FAST TRACK displays the Sort Fields window shown in the following figure:

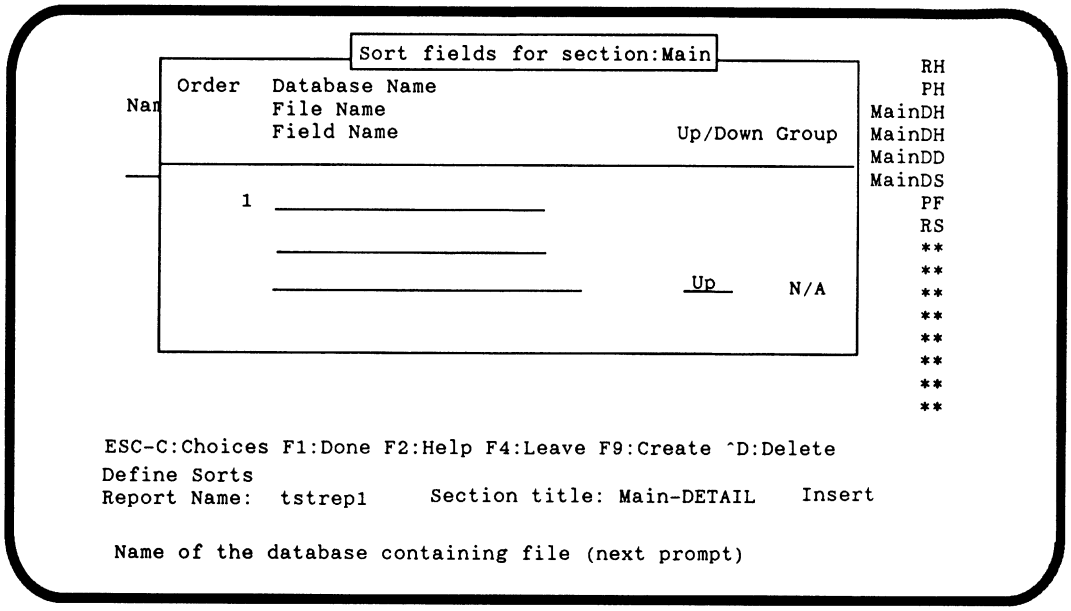


Figure 5-21: Sort Fields Window

FAST TRACK sorts the data alphabetically or numerically depending on the data type of the field by which you choose to sort. You can sort on any field in any file defined for the current report section, or on any variable.

In the Sort Fields window, the Order field displays the default value 1. This indicates that the sort field specified on this line is the primary sort field for the Main section of this report.

In the Database field, enter the name of the database that contains the file.

In the File Name field, enter the filename that contains the field (or **<variable>**) you want to sort. In the Field Name field, specify the sort field, or variable name.

The Up/Down field lets you determine the sort order. Enter **Up** (A–Z) to sort in ascending order. Enter **Down** (Z–A) to sort in descending order. In this particular sort, the Group Name field is not used. Remember, you can use the CHOICES (ESC–C) key to display a list of acceptable values for each field in the Sort field window.

You can sort multiple fields in any number of files. The order in which you enter sort definitions into the Sort Fields window determines the order in which the sorts are evaluated when displaying the report data.

To enter a sort definition after the current sort definition, press [↓] or [RETURN]. Use the [INSERT] (F9) key to insert a new sort definition above the current sort definition.

To delete a sort definition, position the cursor on the sort definition line in the Sort Fields window and press the [DELETE] (F10) key.

When you have finished entering fields for sorting, press [GO] (F1). When you insert data and run your report, the data is sorted according to the sorts that you defined.

5.10 DEFINING INCLUDE FILES

To define an include file in a report, select the **DEFINE→INCLUDES** command from the Report Writer horizontal menu. When you select this command, the following window appears:

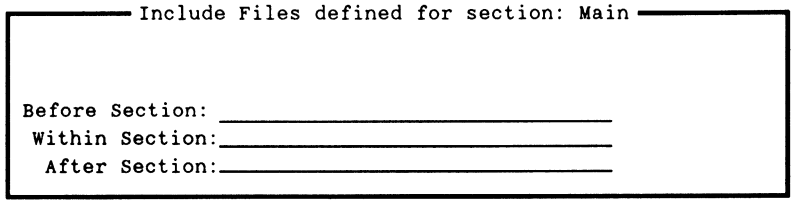


Figure 5–22: Include Files Window

You can use include files to perform complex qualifications or lengthy calculations upon report data. For example, you can use a PROGRESS PROMPT-FOR statement to prompt for a variable used in a qualification.

The Report Writer generates a PROGRESS FOR EACH statement for each section in your report. The input field in which you enter the include file name determines whether the include file is inserted before the PROGRESS FOR EACH statement for the current section, after it, or within it.

To use the Include Files window, enter the name of an existing file (from your current directory) into one of the three section fields, then press the [GO] (F1) key. When you run your report using the COMMAND→VIEW command, FAST TRACK incorporates the include file at the point it is defined in the report.

5.11 DEFINING REPORT SETTINGS

The Report Writer assigns certain default settings for all reports that you create. These settings determine the report width, the page size, the device that receives the output of the report, the directory in which to place report procedures, and who can run your report. The SETTINGS→REPORT command allows you to see these default settings and change them.

When you select the SETTINGS→REPORT command, the Report Settings window appears as follows:

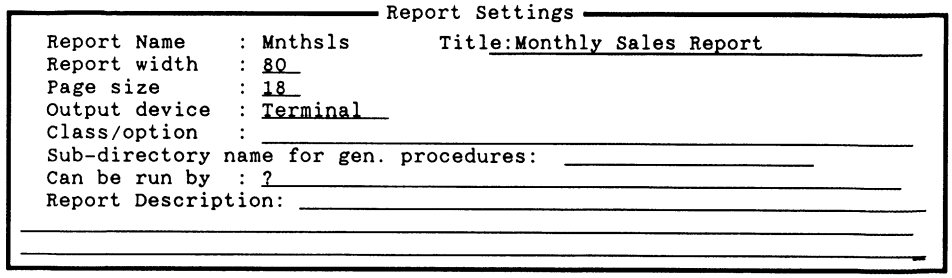


Figure 5–23: The Report Settings Window

This window allows you to update the following report settings:

`Name.` The name of the report.

`Title.` The title can be any string of alphabetic characters including spaces up to 35 characters long. If you entered a report title when you created the current report, the title appears in this field.

`Report Width.` A report can be up to 255 characters wide. Note, however, that most terminals cannot display such wide reports. This width is normally used for output to a printer. The default report width is 80 characters.

`Page Size.` This setting allows you to change the number of lines per page on your report. The default page size is 18 lines.

`Output Device.` This setting allows you to set the output destination of the report output at run time. In other words, this option determines the PROGRESS code in your report procedure that will route the report output when the report is run. You have the following options:

Terminal This value sends the report output to the terminal. If you select this output device, do not specify `Class/option`. This is the default output device.

Printer This value sends the report output to a printer. If you do not specify an output class, the output is sent to a default printer. You can send the report output to a specific printer by naming the printer in the `Class/option` field.

Spool This value sends the report output to a UNIX program and can only be used on UNIX systems. **Spool** can be used to queue files for printing on a specific printer. You must give the name of the UNIX program in the `Class/option` field.

File This value sends the report output to an operating system file in the current directory. You can specify the name of the output file in the

Class/option field. If you do not enter a name in this field, the file receives a default name of *rprtnam*.lis. The default filename consists of the report name (*rprtnam*) and an extension (.lis).

Ask This value lets the user direct the report to an output device at run time. Leave the Class/option field empty. When a user runs the report, a prompt appears asking for the output device and class. The valid device types are: **terminal, printer, spool, file,** and **include.**

Include This value allows you to specify an include file containing PROGRESS code to handle the output routing of the report. You must enter the name of the file in the Class/option field. The file you specify is included in the report procedure in place of the default output statement.

Class/option. Here you can further specify the output device as any one of the following:

printer

spool

file

include

Refer to Chapter 7 for additional information on output classes and options.

Sub-directory for Generated Procedures. This is the name of the directory where you want FAST TRACK to store any FAST TRACK generated procedures.

Can Be Run By. This setting determines who can run the report. The default value for this setting is ? (unknown). The ? value lets you run the current report with both the FAST TRACK and PROGRESS versions of your current database. For more information about access privileges and other security matters, see Chapter 7.

Report Description. This setting is a description of your report. It is used for documentation purposes.

When you finish entering your report settings, press the GO (F1) key to save the settings and return to the Report Writer editing screen. You can change these report settings at any time.

5.12 PAGINATING REPORTS

The Report Writer allows you to paginate your reports based upon the structure on your report. For example, you can paginate on a particular report section. The SETTINGS→SECTION command allows you to set the pagination characteristics for the current report section. The SETTINGS→PAGE_EJECT command allows you to set the pagination information for all of the report sections at once.

When you select the SETTINGS→PAGE-EJECT command, the Page-eject window appears as shown in the following figure:

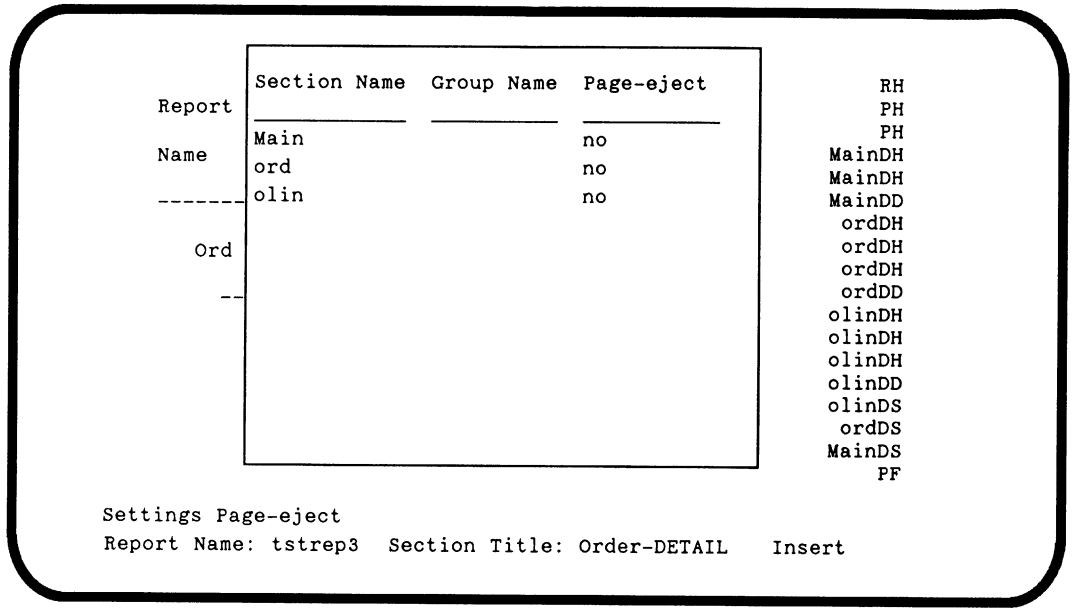


Figure 5-24: Page-eject Window

The Page-eject window displays the section names of all of the sections in your report and lets you change the page-eject setting of any section. If you specify **No** (the default) for a particular section, a page eject does not occur after the section. If you specify **Yes**, a page-eject occurs after the section.

5.13 INSERTING AND MANIPULATING REPORT INFORMATION

During this phase of the report development path, you insert the various items that comprise the information in your report. For example, you can insert and manipulate the following items:

- Field
- Text
- Variables
- Aggregates

To insert or manipulate these items, you must perform some of the actions listed in the definition phase of the development path. For example, you cannot insert fields from a database file into a report until you define the file for use in the report. The following sections introduce the different elements that you can place into a report and a number of commands from the Report Writer horizontal menu that allow you to insert and manipulate the report structure and other aspects of your report.

5.13.1 Fields

After you define the database files from which you want to develop a report, you can insert fields or portions of fields from those database files into your report. Every field consists of two parts as shown in the following diagram:

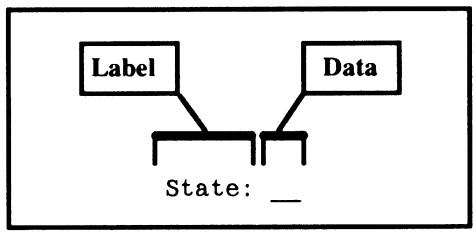


Figure 5–25: Field Structure

The *label* identifies the data and the *data area* displays the data. The Report Writer displays fields in one of two ways:

- With top labels — The labels appear on the line above the data.
- With side labels — The labels appear to the left of the data.

The default manner in which FAST TRACK displays a field differs, depending upon which line in a report section is highlighted when you insert a field into the report.

- If the cursor is located in the report header, page header, main detail summary, page footer, or report summary fields, the default is side labels. If you insert more than one field, a new line is automatically opened for each field that you add.
- If the cursor is located on the detail header, or detail data line, the default is top labels. The labels appear on the detail header line, and the data appears in the detail data line. If the fields do not fit in one row across the screen, FAST TRACK automatically inserts new rows.

You can change the settings for a report section to have either top or side labels for all fields inserted into the section.

There are several commands in the Report Writer horizontal menu that allow you to insert, remove, and manipulate fields in a report. The following sections identify and explain how to use each of these commands.

5.13.2 Inserting Fields, Labels, and Data

Using the Report Writer horizontal menu commands, you can:

- Insert an entire field, including the label and data area.
- Insert only the label of a field.
- Insert only the data area of a field.

The following table presents the commands that allow you to insert all or a portion of a field from a database file into your report.

Table 5-3: Commands to Insert Fields, Labels, and Data Areas

Command	Action
INSERT->FIELD	Insert fields into the current report section.
INSERT->LABEL	Insert field labels into the current report section.
INSERT->DATA	Insert field data areas into the current report section.

You can only insert fields, labels, and data areas from files defined for the current section or those files defined for the parent sections of the current section. For example, if a report section `sec1` is nested in the `Main` section of a report, you can insert fields into the `sec1` section from files defined in both the `Main` and `sec1` sections. However, you cannot insert fields into the `Main` section from files defined for the `sec1` section.

Position the cursor in the report section in which you want to insert fields. When you select any of the commands listed in the table above, a Choices window appears as follows:

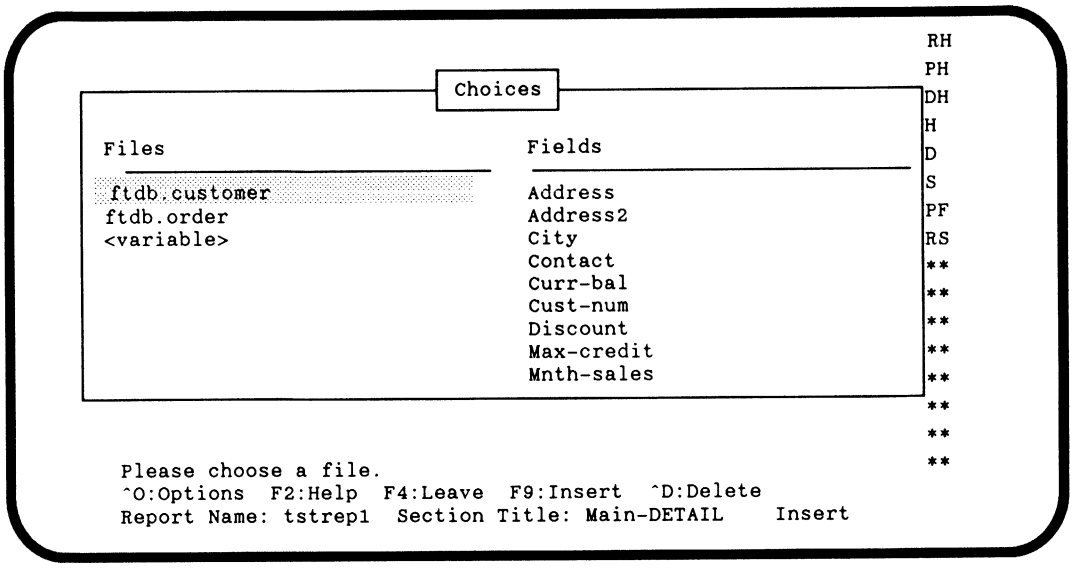


Figure 5-26: File/Field Choices Window

This window lets you select the fields, labels, or data areas to insert into the current report section. There are two columns in the Choices window. The `Files` column contains the names of all the files in the current databases that are accessible by the current report section. The `Fields` column contains the field names of all the fields in the highlighted file in the `Files` column.

To insert a field, label, or data area from a file available to the current report section, you must first select a file from the `files` column. Use the ↓ and ↑ keys to highlight a file name and the press RETURN. The cursor moves to the `fields` column. To select each field, use the ↓ and ↑ keys to highlight a field name and then press RETURN.

The Report Writer places an asterisk (*) next to the selected fields. You can select more than one field. To deselect a field, highlight the field and press RETURN. The asterisk disappears and the field is no longer selected.

Use the ← and → keys to move back and forth between the `files` and `fields` columns and select fields from other files available to the current report section. When you have finished selecting fields, press the GO (F1) key to save your selections and return to the Report Writer editing screen. The fields, labels, or data areas you select appear in the Report Writer editing screen in the order in which you select them. Remember, the INSERT→LABEL command places only the label of the selected fields into the report and the INSERT→DATA command places only the data area of the selected fields into the report.

5.13.3 Moving Fields, Labels, and Data

The Report Writer horizontal menu has several commands that allow you to move fields, labels, and data areas on your report. The PICK commands allow you to mark and move fields and areas on your screen.

You can move a field to a new report section only if the field's database file is available for use within the new report section. FAST TRACK does not perform a validation if a field is valid in the new position. If you move fields between sections, ensure that the fields are valid in the new section. Otherwise, FAST TRACK generates a compilation error message when you attempt to view or generate the function.

The following table presents the PICK commands, which allow you move fields, labels, data areas, and groups of fields in a report.

Table 5-4: Pick Commands

Command	Action
PICK->FIELD	Select the field at the current cursor location for an operation.
PICK->OBJECT	Select the label, data area, or text string at the current cursor location for an operation.
PICK->AREA	Select a group of fields for an operation.
PICK->MOVE	Move the currently ''picked" field, object, or area to the current cursor location.
PICK->UNDO	Unpick the currently ''picked" field, object, or area.

For information about how to use these commands, see Appendix C.

5.13.4 Setting Field Display Characteristics

Just as you can change the default report and section settings, the Report Writer allows you change the default settings of fields in a report. The default settings of a field are set in the schema of the current database.

The SETTINGS→FIELD command allows you to change the label and the data format of a field in your report. Before selecting this command, position the cursor in the field for which you want to change the defaults. When you select this command, the Field Attribute Setting window appears as follows:

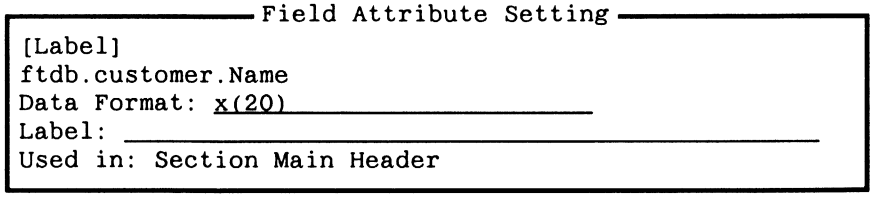


Figure 5-27: The Field Attribute Setting Window

At the top of the window, FAST TRACK displays the database file and field label of the current field. The section in your report where the field is used appears at the bottom of the window. You can change two settings for the current field:

- The data display format
- The field label

For additional information on data display formats, refer to the *Programming Handbook*.

After you change the label and data format settings, press GO (F1) to enter the changes and return to the Report Writer editing screen. Changes made with this command affect only the display characteristics of a field on your report, not the schema of your current database.

5.13.5 Removing Fields, Labels, and Data

The Report Writer supplies several commands in the Report Writer horizontal menu that allow you to delete fields, labels, and data from a report. The REMOVE→FIELD command removes the field at the current cursor location. Before selecting this command, position the cursor in the field that you want to delete. This command removes the entire field, including the label and the data area. To remove more than one field at a time, or to remove only a field label or data area, use the REMOVE→PICKED command.

The REMOVE→PICKED command removes the current field, object, or area highlighted with one of the PICK commands. Before selecting the REMOVE→PICKED command, use the PICK→OBJECT, PICK→AREA, or the PICK→FIELD command to pick a field, label, data area or a group of fields. Once a field, object, or area has been picked, select the REMOVE→PICKED command to remove the item.

5.13.6 Text

You can type text into any section in a report. Simply, position the cursor where you want the text to begin and start typing. You cannot enter text into your report when the Report Writer horizontal menu is displayed. Use the BACKSPACE and DEL keys to edit any text strings or labels that you type into your report.

The SETTINGS→MODE→INSERT/OVERSTRIKE command allows you to toggle between insert and overstrike mode.

In insert mode, FAST TRACK displays the mode setting in the lower right corner of your screen. The character you type is inserted at the current cursor position. The cursor and the character under it move one space to the right.

In overstrike mode, the character you enter *replaces* the character at the position of the cursor. The [MODE] (F3) key performs the same function as the SETTINGS→MODE→INSERT/OVERSTRIKE command

5.13.7 Moving and Removing Text

Once you have inserted a text string into a report, the Report Writer allows you to move it to another position or delete it. Field labels are essentially text strings. Therefore, you can use the same commands to move and remove text strings that you use to complete the same operations on labels.

To move a text string to a different position in a report, position the cursor on the text string and then select PICK→OBJECT command. This command highlights all the text in the current row. To pick only a portion of the text in the current row, use the PICK→AREA command. After you pick the text, position the cursor at the new location and then select the PICK→MOVE command to move the text string to the new location.

To remove a text string from a report, position the cursor on the text string and then select the PICK→OBJECT or PICK→AREA command. Next, select the REMOVE→PICKED command to remove the text string.

5.13.8 Variables

You can define a variable as a means to temporarily store and display calculated data in a report. The data stored in a variable is not part of your database, but may be based upon calculations using data from your database or elsewhere. To use a variable:

- Invoke the DEFINE→VARIABLES command to define the variable.
- Invoke the INSERT→FIELD to insert the variable into your report.

Both of these commands are available on the Report Writer horizontal menu. See the section 5.7 for information about defining variables for use in a report. The following sections tell you how to insert a variable into a report.

5.13.9 Inserting Variables

Once you define a variable for a report section, you can use the INSERT→FIELD command to insert the variable into the report. Before selecting this command, position the cursor in the report section for which you defined the variable. When you select the INSERT→FIELD command, the following window appears:

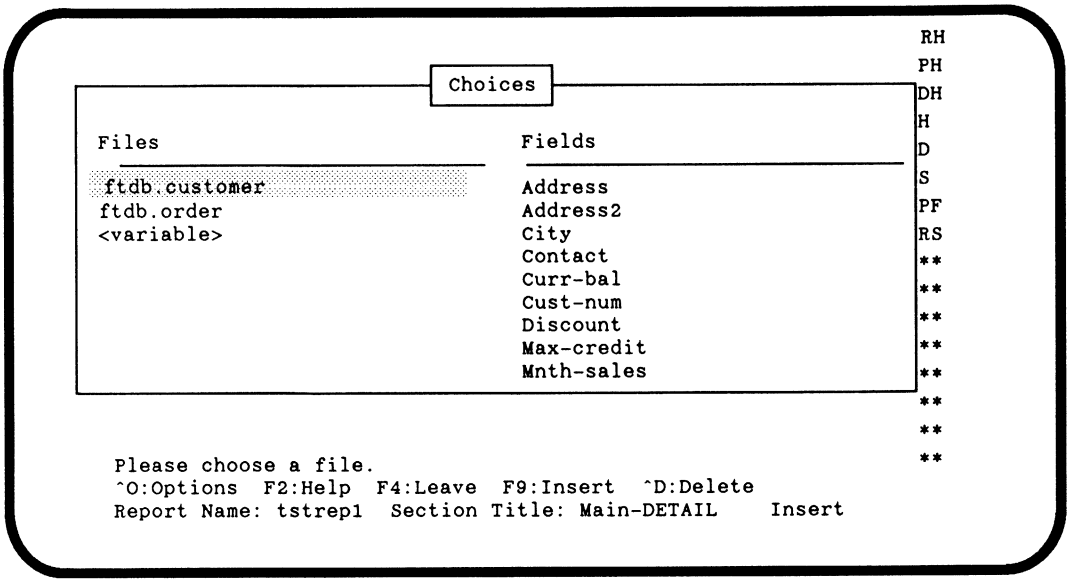


Figure 5-28: Field/File Choices Window

To insert a variable, select the <variable> entry in the Files column. Use the ↓ and ↑ keys to highlight the entry then press RETURN. The variables defined for the current report section appear in the Fields column. The cursor is on the first variable in the fields column.

To select a variable to insert into the report, highlight the variable name and then press RETURN. You can insert as many variables as are defined in this column. You can also define and insert a new variable or edit an existing definition by selecting the <new variable> option in the Fields column. When you select <new variable>, the Define Variables window appears and you can define variables as explained in the section "Defining Variables."

When you finish selecting variables, press the GO (F1) key to insert the variables in the current report section at the current cursor location. The Report Writer places both the label and the data area of the variable into the report.

5.13.10 Manipulating Variables

Once you have inserted a variable into a report, the Report Writer treats the variable as a field. This lets you use the same commands to alter variable field settings, and move or remove the variables you use to complete the same operations on fields.

For more information about moving and removing fields in a report, see sections 5.13.3 and 5.13.5.

5.13.11 Aggregates

The Report Writer supplies several useful tools that allow you to perform calculations to summarize data and other useful functions such as page numbering. The INSERT→VALUE command accesses these tools. This command is commonly used in the header, footer, and summary report sections.

The INSERT→VALUE command allows you to insert a calculated value into your report at the position of the cursor. Because this command does not insert a label into your report, you need to type in a label of your choice, if you want to identify the calculated value.

The Choices window containing valid choices for aggregates appears when you invoke the INSERT→VALUE command. Figure 5-29 shows the Choices window in this situation.

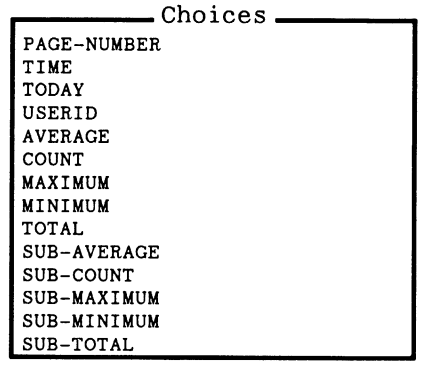


Figure 5-29: Choice Window for Aggregates

Use the [↓] and [↑] keys to scroll through the choices. The values that you can insert are as follows:

PAGE-NUMBER	Inserts a page number at the current cursor location.
TIME	Inserts the time of day at the current cursor location.
TODAY	Insert the date at the current cursor location.
USERID	Inserts the ID of the user who is logged in at the time the report is created. The ID appears at the current cursor location.

Most of the values that you can insert for an aggregate are numerical. For instance, you might want to display the total of the data in a given field for all of the records in your report. When you enter an aggregate operation, an operation window appears. Figure 5-30 shows a typical operation window.

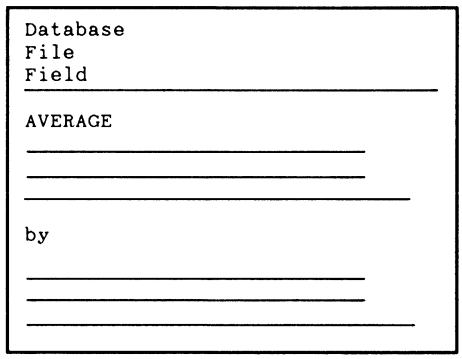


Figure 5-30: Window for the Average Value

This window is for the AVERAGE value. You must enter the name of the file and the field that you want to use to calculate the aggregate. If you want to accumulate an aggregate by break group, enter the file and field of the break as a *by* clause. This situation arises, for example, if you want to display the average current balance for each sales region (within the customer file).

AVERAGE	Calculates the average of a set of data.
COUNT	Tells you how many entries there are in your current group.
MAXIMUM	Returns the maximum value for a given field in the database. It can be used with any data type.
MINIMUM	Returns the minimum value for a given field in the database. It can be used with any data type.
TOTAL	Returns the total of the field value for the entire database or break group. You can only use this aggregate function with numeric data types.
SUB-AVERAGE	Returns the average of the value of the field in a break group. You can only use this aggregate function with numeric data types.
SUB-COUNT	Returns the number of times the value of a field has been counted in a break group. You can only use this aggregate function with numeric data types.
SUB-MAXIMUM	Returns the maximum value of a field in a break group. You can only use this aggregate function with numeric data types.
SUB-MINIMUM	Returns the minimum value of a field in a break group. It does not supply a minimum value for all records, just those in a break group. You can only use this aggregate function with numeric data types.

SUB-TOTAL Returns the sub-total of the value of the field in a break group. You can only use this aggregate function with numeric data types.

When you finish entering the aggregate information into the window, press GO (F1) to enter the aggregate at the current cursor location in the report Writer editing screen.

5.13.12 Manipulating Aggregates

Once you have inserted an aggregate into a report, the Report Writer treats the aggregate as a data area. Therefore, you can use the same commands to manipulate aggregates that you use to complete the same operations on data areas.

For more information about moving and removing fields in a report, see the previous sections entitled "Moving Fields, Labels, and Data," and "Removing Fields, Labels, and Data" in this chapter.

5.14 VIEWING REPORT OUTPUT

After you finish defining your report structure and filling in the structure with text, fields, labels, and data, it is a good idea to view your report before saving it to your database or generating a PROGRESS procedure for the report. Use the COMMAND→VIEW command to generate, compile, and run a temporary report procedure and display the report output on your terminal. You can scroll through your report by pressing SPACEBAR. After you view a report, FAST TRACK returns the cursor to the Report Writer editing screen.

5.15 GENERATING REPORT PROCEDURES

Select the COMMAND→GENERATE command to create a PROGRESS procedure for your current report. FAST TRACK displays a message asking if you want the database prefix on the file names. If you accept the default value of *yes*, the database name will prefix the file names for the generated procedures. This is necessary if you have the same filename in different databases, but it also requires that you run this procedure against a database with this logical name or alias. If you know that the filenames are unique across all connected databases, you can generate the procedure without the prefix and you can run the procedure against any database configuration.

While FAST TRACK compiles the report, you see these messages on your screen:

```
Now writing your report program ...

Your report program is written, now compiling...
```

When FAST TRACK has finished compiling, it creates a both a procedure (.p) file and compiled (.r) file for the report, and places them in the current directory.

The compiled file is important because it contains the *object* code that you distribute as part of your finished application. For more information about compiled files, refer to Chapter 7 and Appendix D.

Along with procedure and compiled files, the Report Writer generates an include file for every section in your report. The Report Writer appends a section number to the end of the report name and places a `.i` extension after the section number.

For every report for which you generate a procedure, FAST TRACK generates at least one `.i` file. When you run the report procedure, FAST TRACK uses these include files to format the data in a particular report section.

Procedures generated with the Report Writer can be used as *include files* in other PROGRESS procedures and they can also be edited using the PROGRESS editor. For more information about include files, see the *PROGRESS Language Reference* manual.

Note that the COMMAND→GENERATE command does not save your report as an object in the current database.

5.16 TESTING REPORT PROCEDURES

You can test your report using the OTHER→GOTO command. When you select this option, FAST TRACK displays the following window.

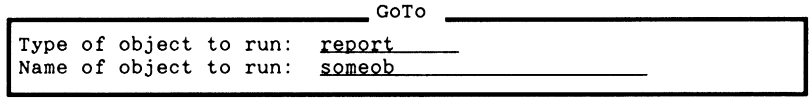


Figure 5–31: The GOTO Window

Ensure that you enter the correct name of your report — that is, the name you gave the report in the Report Writer initialization window — in the `Name of object to run` field. When you use the OTHER→GOTO option, ensure that the report has been generated.

5.17 SAVING REPORTS AS DATABASE OBJECTS

When you save your report using the COMMAND→SAVE command from the Report Writer horizontal menu, the Report Writer saves your report as an *object* in the current database.

In other words, when you save your report to the current database all the section, file, field, and formatting information for the report is saved in various tables in the schema of your current database.

After you save a report as an object in your current database, you can retrieve the object to recreate your report or use it as a template for other reports.

You can also delete objects from your current database, using the `Maintenance` option on the FAST TRACK main menu. See Chapter 8 for additional information on deleting objects.

The LEAVE→SAVE command also saves a report as an object in the current database. When you select the LEAVE→SAVE command, the Report Writer returns you to the FAST TRACK main menu and ends the current report editing session.

5.18 LEAVING THE REPORT WRITER

The LEAVE commands allow you to leave the Report Writer and return to the FAST TRACK main menu. The LEAVE→SAVE command saves your report as an object in the current database and returns you to the FAST TRACK main menu. The LEAVE→QUIT command returns you to the FAST TRACK main menu without saving the changes that you made to the current report since the last time you saved the report.

Chapter 6
The QBF Generator

A *QBF* (query-by-form) is a procedure that displays selected data in a predetermined format. It allows you to find, examine, and update the records in a database by filling in a form.

A QBF can provide your end-user application with a flexible, yet expedient, method of querying the database. Additionally, FAST TRACK's QBF provides capabilities to join, add, and delete database records.

This chapter provides the following information about using the QBF Generator:

- Why use a QBF?
- Development path for creating a QBF.
- Creating a QBF.
- Generating a QBF.
- Running a QBF.
- QBF command options.
- The JOIN command.
- The QUERY command.
- QBF command summary.

6.1 WHY USE A QBF?

The QBF Generator creates procedure files you can use for basic file maintenance, saving you the time of having to write the equivalent PROGRESS statements for query, join, add, and delete operations.

The form determines what file information is entered or retrieved and the organization of the information on the screen. A QBF procedure can perform the following actions:

- Look at individual database records, move forward and backward through the database file one record at a time, or go directly to a specific record.
- Find and view related records from several files.
- Search for a subset of records in the database file. You select a subset of records by defining search criteria that a record must pass in order to be included in the subset.
- Perform database file maintenance operations on the records in the form. These operations include adding, deleting, updating, and printing records.
- Provide scrolling capabilities if you generate it against a down form.

Figure 6-1 shows a typical QBF form, adapted from the customer file in the demo database.

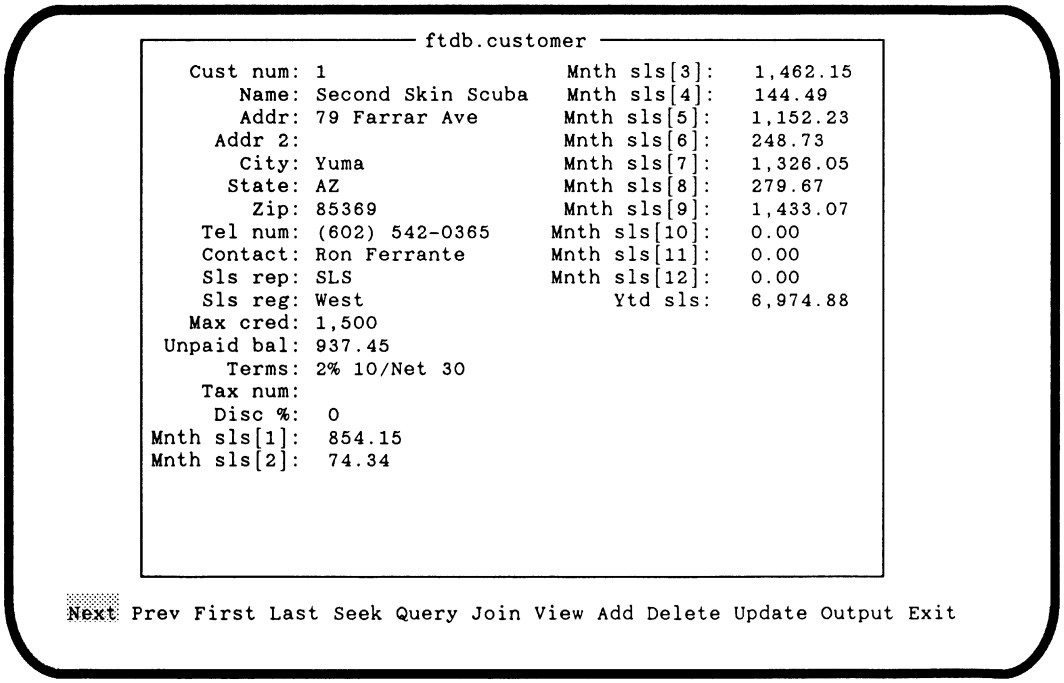


Figure 6-1: Typical QBF Form

The QBF Generator features an extensive set of commands that the end-user can access to perform different QBF functions. You issue QBF commands by using its horizontal command menu. Table 6-1 lists the horizontal menu commands and their associated function.

Table 6–1: QBF Command Menu Options

Option	Action
NEXT	Displays the next record in the file.
PREV	Displays the previous record in the file.
FIRST	Displays the first record in the file.
LAST	Displays the last record in the file.
SEEK	Displays a record based on index value you specify.
QUERY	Finds records that match the qualifications you specify.
JOIN	Joins two related files.
VIEW	Displays available QBF forms.
ADD	Adds a record to the file.
DELETE	Deletes a record from the file.
UPDATE	Lets you edit and update the current record.
OUTPUT	Sends the current record to a specified output device.
EXIT	Exits the current process.

NOTE: Some of these options may not be available. See "DBRESTRICTIONS" in the *PROGRESS Language Reference*.

6.2 DEVELOPMENT PATH

Before you create a QBF, determine whether you require a standalone QBF, or one linked to the Menu Editor or Screen Painter. The following list provides a typical sequence for creating, specifying, generating, and running a QBF.

Choose the files for which you want to run the QBF.

Specify database generation operations for the QBF.

Generate the QBF.

6.3 CREATING A QBF

To begin the process of creating a QBF, choose the QBF Generator option from the FAST TRACK Main Menu. You can also create a QBF from the Screen Painter. You should weigh the consequences of where you create a QBF, however. Here are some guidelines:

- You can only create a QBF for one form at a time using the Screen Painter or Menu Editor.
- The QBF Generator allows you to create several QBFs at once, and tie them together with a QBF menu procedure.
- FAST TRACK does not create a QBF menu procedure when you create a QBF using either the Menu Editor or the Screen Painter.

As these guidelines indicate, creating a QBF in the QBF Generator can be expedient. If you want to customize your QBF output, however, invoke the QBF function via the Menu Editor or Screen Painter. Table 6–1 describes the QBF menu commands.

6.3.1 Creating a QBF from the Screen Painter

You can use the Screen Painter to create a QBF by accessing the Screen Painter from the Main Menu, or accessing it via the Menu Editor.

Once you have developed and tested a screen in the Screen Painter, press OPTIONS (CTRL-O) and select the DEFINE→QBF option. At this point, FAST TRACK displays the screen shown in Figure 6–2.

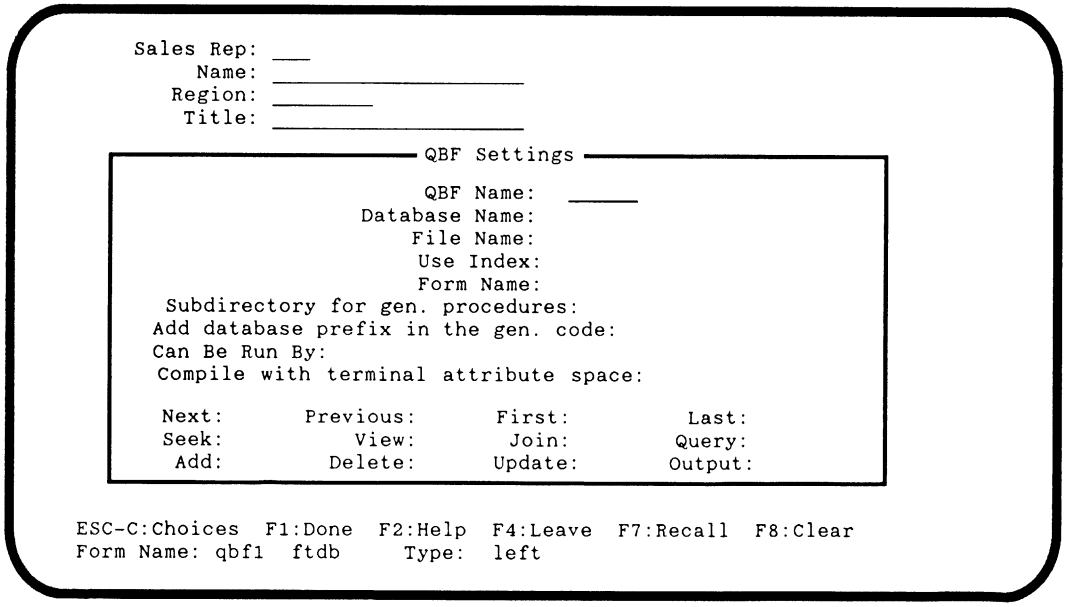


Figure 6–2: QBF Settings Window Accessed from the Screen Painter

The QBF Settings window lets you set the characteristics and capabilities of the QBF operation associated with your current Screen Painter form. The following sections explain the fields in the QBF Settings window.

QBF Name. Enter a QBF name up to seven characters in length. FAST TRACK uses this name to identify your QBF procedure as well as the QBF in the FAST TRACK database. Note that you can only use alphanumeric characters and the underscore symbol.

Database Name. This is the name of the database that contains the file indicated in the File Name field.

File Name. This is the name of the database file for which you want to generate the QBF. You cannot change this value, because it is determined by the input file for the current form. In this example, the input file is the `salesrep` file.

Use Index. The QBF always uses an index to read records, and the index definition determines the order in which you will see the records. By default, the primary index is used. If the database file has other indexes, one of these can be used instead. You can press the CHOICES (ESC C) key when the cursor is in this field, to see a list of valid indexes for the current file.

Form Name. The name of the form used by the QBF. The default form name is the same as the QBF name. You can name the form anything you like.

Sub-directory Name for the Gen. Procedure. This requests the name of the subdirectory for your QBF. If you do not specify a directory name here, FAST TRACK puts the QBF in your current working directory by default.

Add Database Prefix in the Gen. Code. Unless you have a file with the same name in another database, you may answer **no**. If you answer **yes**, the procedure will only run for a database that has the same name as the one specified in the Database Name field.

Can Be Run By. A list of users who can access the QBF. By default, anyone can use the QBF. For more information about PROGRESS security, see Chapter 11 of the *Programming Handbook*.

Compile with terminal attribute space. If you need to develop the QBF for a spacetaking terminal, specify **yes** to this item. The default depends on the nature of the terminal. (See Chapter 7 of the *Programming Handbook* for more information on space-taking terminals.)

You then have the option of choosing which options will appear on the command menu of the QBF procedure (`ADD`, `UPDATE`, etc.)

6.3.2 Creating a QBF for a Down Frame

The Screen Painter provides a way to produce a QBF for a down frame. Use the Screen Settings window to set `Form Type` to **top**. At `Repeat form` down, specify how many lines down you want the form to show at a time. For instance, if you specify **6**, then the form will show six lines down.

Next, insert fields, then generate the QBF. You can scroll through the data on the screen using the ↓ , ↑ , RETURN , page-up and page-down keys.

6.3.3 Creating a QBF from the QBF Generator

The QBF Generator provides a quick way to produce default QBFs for files in a database. When you choose the QBF Generator option from the Main Menu, the following screen appears (if you are using the demonstration database).

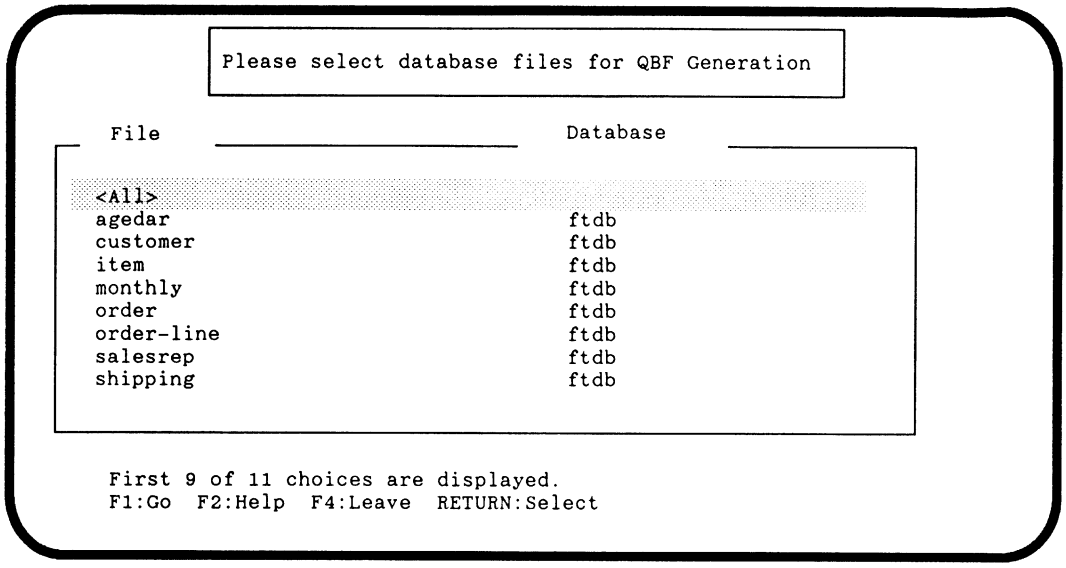


Figure 6–3: The QBF Generation Database File Choices Window

Before FAST TRACK can create your QBFs, you must first chose all the files for which you want to generate a QBF. Then you need to provide the information requested in the Database QBF Generation Options screen. Note that if a file has more than one screen defined for it, a QBF is generated for each form.

To choose a file, highlight the filename and press RETURN. If you make a mistake, you can unmark a filename by highlighting it again and pressing RETURN. When you have marked all of the files that you want, press GO (F1).

After you have marked and chosen your files, the list of choices disappears and the cursor moves to the QBF Generation Options window shown in the following figure.

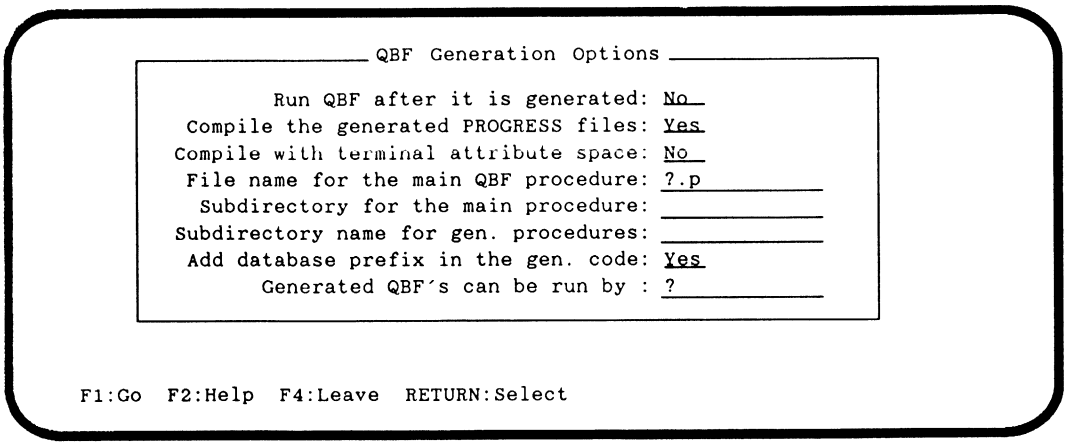


Figure 6–4: The QBF Generation Options Window

You need to specify the following QBF generation options:

Run QBF after it is generated. If you answer *Yes* to this option, FAST TRACK generates the QBF and a main QBF procedure, and then immediately runs the main procedure. If you specify *No,* FAST TRACK returns to the Main Menu after generating your QBFs.

Compile the generated PROGRESS files. This option allows you to specify whether you want FAST TRACK to compile the forms and QBF procedures that are generated. A compiled QBF starts executing more quickly than the uncompiled version. Alternatively, you might just want to generate the procedures now and compile them later, if for instance memory is at a premium. You can compile them using the Maintenance Menu.

Compile with terminal attribute space. If you expect the QBF to be run on a space taking terminal, you should specify **Yes** to this item. (See Chapter 7 of the *Programming Handbook* for more information on spacetaking terminals.)

File name for the main QBF procedure. The QBF Generator creates a main menu procedure which ties together all of the QBFs generated. You can run a QBF procedure independently; or alternatively, you can run a QBF by selecting it from this main menu. This item in the QBF Database Generation Options window requests that you enter the name for this QBF procedure

Sub-directory for the main procedure. This option allows you to specify the subdirectory in which you want the main QBF procedure to be located. If you do not specify a name here, FAST TRACK places the file in your current working directory.

Sub-directory name for the gen. procedures. This option allows you to specify the sub-directory in which your generated QBF procedures will be located. If you do not specify a directory name here, FAST TRACK puts the QBF in a subdirectory named after the database to which the file belongs.

Add database prefix in the gen. code. This option allows you to specify whether or not the database prefix will be added to the generated code.

Can Be Run By. This setting determines who can run the QBF menu procedure. The default value for this setting is ? (unknown). This means that there is no FAST TRACK security code in the procedure. For more information about access privileges and other security matters, see Chapter 7.

6.4 GENERATING A QBF

After you have finished entering all of the necessary information in the QBF Generation Options window, press GO (F1) or RETURN to generate your QBFs.

Now the QBF Generator goes to work. FAST TRACK first looks for all forms in the database using the files you have chosen to create the QBF. If a form is found, FAST TRACK generates a QBF procedure for it. If there is more than one form for the file, FAST TRACK generates a QBF procedure for each form. If no form is found for the file, FAST TRACK automatically produces a default form.

FAST TRACK also stores any other files associated with the QBF in the subdirectory you specified in the QBF Generation Options window. As it works, the QBF Generator informs you of its actions by displaying messages as it creates default forms for files, generates QBFs for forms, and compiles the QBF procedures.

For each form in the previous example, FAST TRACK generates a procedure (.p) file, which you can run from the PROGRESS editor. After the QBF generator has generated all of the QBF forms and procedures, it generates a QBF Menu procedure that you use to run your QBF. These files can only be run with the database from which they were created or a copy of that database. Refer to Appendix D for additional information on generated QBF files.

If you specified **No** to the `Run QBF after it is generated` option, FAST TRACK exits to the Main Menu when the QBF Generator finishes. Otherwise, it displays a menu containing all generated QBFs.

6.5 RUNNING A QBF

There are several ways for you to run your QBF:

- Run the QBF immediately after generating it.
- Define your QBF as a menu choice action using the Menu Editor.
- Run the QBF procedure from the PROGRESS editor.
- Choose the OTHER→GOTO option from the horizontal menu,and specify QBF as the object type and the QBF name as the object name.

6.5.1 The Generate and Run Option

If you specified Yes to the Run QBF after it is generated option, FAST TRACK runs your QBF immediately after generating it. In this case, FAST TRACK displays a QBF menu similar to the one in the following figure.

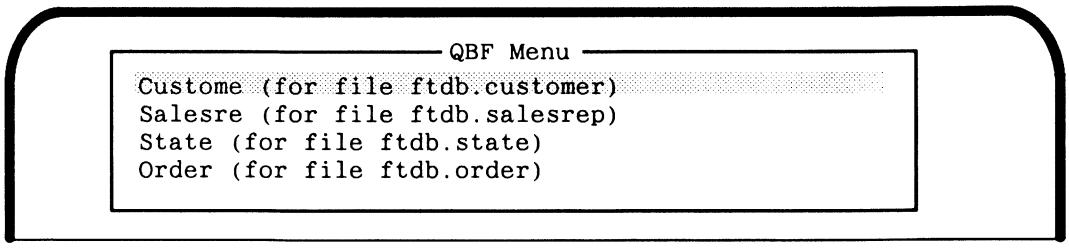


Figure 6–5: Menu of QBF Procedures

To run a QBF, you simply move the cursor to highlight it, and then press RETURN or GO (F1).

6.5.2 Tying a QBF to a Menu Choice

When you are defining menus in the Menu Editor, you can define a menu choice as a QBF. For example, if you have a form named myform and an associated QBF with the same name, you can tie it to a menu as a procedure choice (option). In this case, the appropriate name is `myform.p`. Alternatively, if you tie it to a menu as a QBF choice (option), the appropriate name is simply `myform`. If you generate the QBF from the Screen Painter of the Menu Editor, you may give the QBF a name other than the form name.

6.5.3 Running a QBF from the PROGRESS Editor

To run your QBF Menu procedures from the PROGRESS editor, type the following:

```
RUN yourqbf.p.
```

where *yourqbf* is actually a QBF name. After you have typed this line, press GO (F1).

NOTE: When running a QBF, you may get an error message if, for instance, you try to add a duplicate record. In such a case, FAST TRACK displays an error message with the following: "dbname$filename ..." or "dbname_filename ...", where the first part is the database name, the second part is the filename, and the "$" or "_" is a separator.

6.6 QBF COMMAND OPTIONS

As noted in the previous sections, you can run a QBF from either FAST TRACK or the PROGRESS editor. Once you have run your QBF, a menu of QBF commands appears at the bottom of the screen. The following figure shows this menu.

```
Next Prev First Last Seek Query Join View Add Delete Update Output Exit
```

Figure 6–6: QBF Command Menu

The JOIN and QUERY commands are the two most powerful commands on this menu. They allow you to view files from restricted subsets of records in your file. The following sections explain the JOIN and QUERY commands as well as other QBF features that assist you in manipulating QBF data.

6.7 THE JOIN COMMAND

The JOIN command allows you to display a relationship between two files in the current database. The two files must have a corresponding QBF procedure and the files must be related by a unique index define in the PROGRESS Data Dictionary. When you use the JOIN command in a QBF procedure generated with the QBF Generator, you can only relate two files if they are represented by a corresponding QBF procedure on the same QBF menu. When you use the JOIN command in a QBF procedure generated in the Screen Painter or Menu Editor, you can related any two files if they have a corresponding QBF procedure and an indexed relationship defined in the current database.

A QBF procedure establishes a *parent–child* relationship between files. A *parent* file is any file previously defined in the current QBF that is related to the current input file by a unique index in the database schema. In other words, all index fields in a unique index in one of the files must exist in the other file. They must also have the same name and data type.

If there is more than one unique index relating the parent to the current input file, you can choose the index you want to use from the QBF Menu, which is illustrated in the following figure.

```
QBF Menu
Order (for file ftdb.order)
Salesre (for file ftdb.salesrep)
State (for file ftdb.state)
```

Figure 6–7: The QBF Menu Window

When you make a selection from this menu, the appropriate QBF overlays the record you are viewing. The following figure shows an overlay screen:

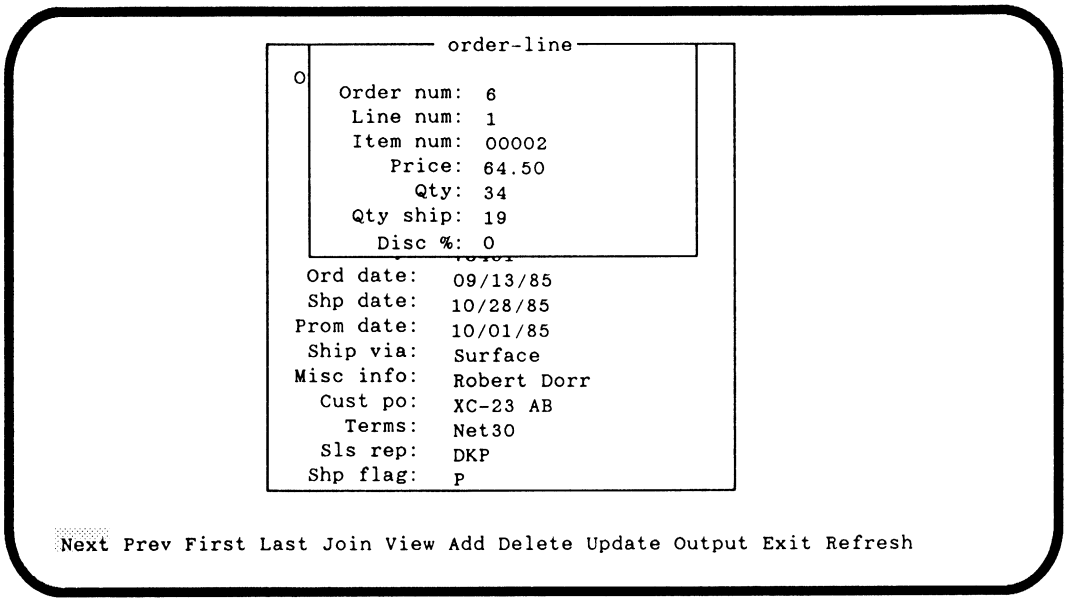


Figure 6-8: The Order-line QBF in Overlay Format

In this example, the overlay window displays joined data associated with the original record. After selecting the `order-line` QBF, and pressing RETURN, your QBF displays all of the line numbers for your order.

6.8 THE QUERY COMMAND

You use the **QUERY** command to select only those records that meet certain criteria. This command allows you to retrieve and process records that meet the selection criteria you specify.

Once the query has collected the subset of records, you can perform a limited set of actions on the records. That is, you can browse through the subset with the NEXT, PREV, FIRST, VIEW, and LAST commands. And, you can ADD, DELETE, UPDATE, and OUTPUT individual records in the subset. When you select the QUERY command, two windows appear overlapping the record shown prior to invoking the command. The following figure shows a sample query screen.

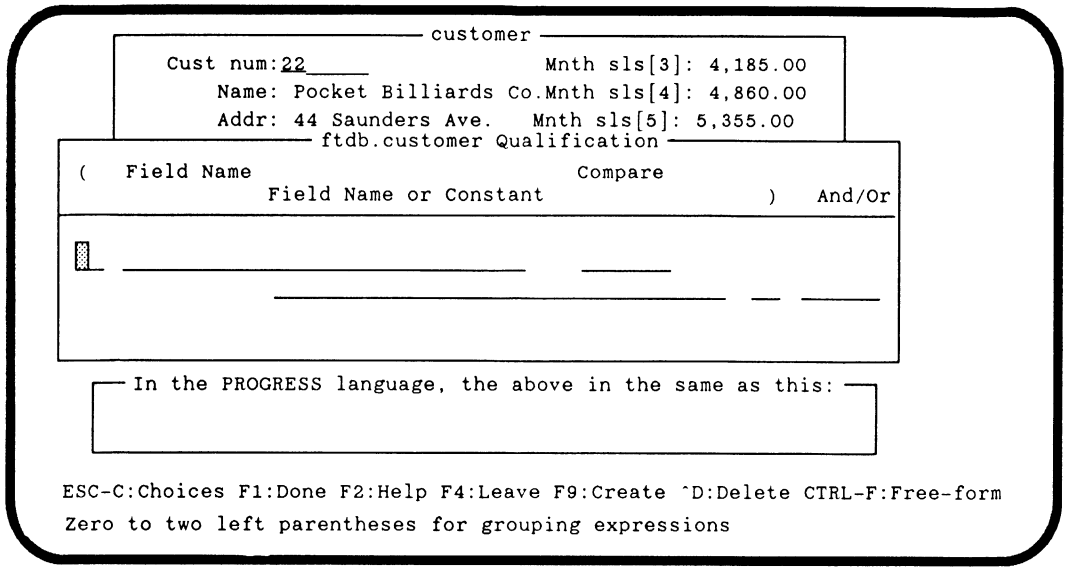


Figure 6–9: The Customer Qualifications Window

The two windows are the Qualification window and the Language window. You specify the basis of your qualified search in the qualification window. As you do so, FAST TRACK displays its PROGRESS language translation in the Language window. Every line in the qualification window represents a PROGRESS clause. At the bottom of the screen is a help message for each field in the Qualification window.

6.8.1 Entering a Query

A query is built from one or more comparison expressions. A comparison expression has three parts:

- The name of the field that you want to test.
- A comparison operator that states the kind of test that should be performed.
- The name of another field or a constant value against which the field is compared.

The format for a query qualification is given in the top line of the Qualification Window. The Qualification window has six fields:

Left Parenthesis. Denoted by (, this field accepts zero to two parenthesis. Parenthesis are optional. If used, a right parenthesis must be paired with a left parenthesis.

Field Name. You enter the name of the field upon which you want to base your restricted search in this field.

Compare. You enter a comparison operator in this field.

Field Name or Constant. This is the name of a field or a constant expression., used when a value is compared to the first field.

NOTE: When you enter a constant date, you should always enter it in the order month, day, then year, even if the display format is different. Also, constant logicals should always be entered as yes/no or true/false, not the format values. For example, if the fomat of field shipped is shipped/not shipped, you wouldn't want to check for shipped = "shipped", but you would want to check for shipped = true.

Right Parenthesis. Denoted by), this field accepts zero to two parenthesis. Parenthesis are optional. If used, a left parenthesis must be paired with a right parenthesis.

And/Or. A logical connective that combines qualification expressions.

Here's a query based on typical information supplied in the Qualification window (the query asks for all records for which the state is Massachusetts):

```
(st EQ "MA")
```

Note that for simple one-clause queries such as this, the parentheses are not required. The parentheses are used to clarify more complex, multiple-clause queries.

6.8.2 Moving Around in the Qualification Window

Enter a qualification in the Qualification window by directly entering the data on the qualification lines, then press [RETURN] or [TAB] . You can also use the [→], [←], [BACKSPACE] , and [DEL] to move the cursor a character at a time and edit mistakes.

After making an entry in the `Field Name`, and pressing [RETURN] , the QBF automatically fills in the `Compare` field with a default comparison operator for the particular data type of the field. These defaults are as follows:

- For integer, decimal, and date data types, the default is `EQ`.
- For the character data type the default is `BEGINS`.

You can press [CHOICES] ([ESC] [C]) with your cursor in any field of the Qualification window to see a menu of available choices. Making a selection from this menu and pressing [RETURN] , enters the choice automatically. The following figure shows the Choices window with the available choices for the `Compare` field.

6.8.4 Logical Operators

More complex qualifications can be constructed any of several ways. One is by combining several simple comparison expressions with AND, AND NOT, OR, or OR NOT. You enter each expression in a connected qualification on a separate line in the Qualifications window.

The following figure shows the Qualification window and an example qualification using the item file from the demo database. (Subsequent examples also use this file).

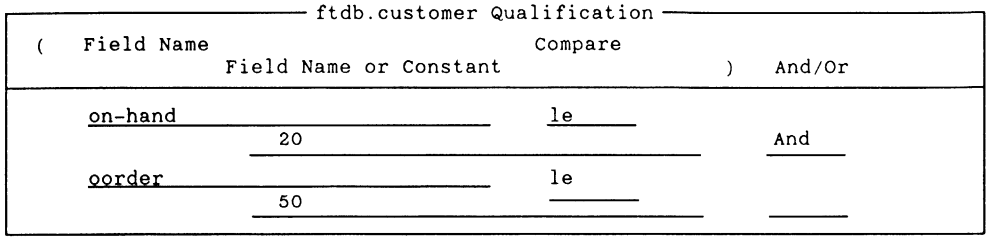


Figure 6-11: Qualification Window

After entering the And/Or expression at the end of the line, you move to the next line by pressing ↓. The following sections explain the And/Or operators.

AND. Returns a true result when both expressions are true. Thus, both expressions must be true for a record to be included in the subset as in the following example.

```
(on-hand le 20) and (oorder le 50)
```

In this example, the qualification is true if items on-hand are less than equal to 20 *and* items on-order (oorder) are less than or equal to 50.

OR. Returns a true result when either expression is true, or when both expressions are true. The following example is valid.

```
(on-hand eq 0) or (oorder eq 0)
```

In this example, the qualification is true if either items on-hand *or* items on-order (oorder) are equal to 0.

NOT. Negates the value of the subsequent expression. It is always used with one of the logical connectives AND or OR as in the following example.

```
(on-hand le 20) and not (oorder gt 50)
```

In this example, the qualification is true if items on-hand are less than equal to 20 *and* items on-order (oorder) are not greater than 50. Observe that this example produces the same result as the example used for the AND operator.

6.8.5 Order of Evaluation

FAST TRACK evaluates logical expressions exactly the same way as PROGRESS does. In compound expressions, the rule of thumb is AND always dominates OR. In other words, FAST TRACK always evaluates expressions joined by AND before it evaluates expressions joined by OR.

For example, if you are looking for those customers who live either in Boston, Massachusetts or in San Diego, where San Diego can be in any state, then parentheses can be added to make this meaning clear:

```
(st EQ "TX" AND city EQ "El Paso") OR city EQ "Houston"
```

If you are looking for customers who live in El Paso or Houston, where either city must be in Texas, then parentheses are absolutely necessary to clarify the meaning. For example:

```
st EQ "TX" AND (city EQ "El Paso" OR city EQ "Houston")
```

Using this method, your qualifications can be quite complex. Additionally, you can use free-form expressions for even more complex expressions.

Refer to the *PROGRESS Language Reference* manual for additional information on the precedence of logical operators.

6.8.6 Free Form Qualifications

Free form qualifications allow much more freedom in specifying a qualification than is possible with the structured Qualifications window. When the cursor is in the Qualification window, press CTRL-F. The following figure shows the Free Form Qualification window as it would appear for the customer file from the demo database.

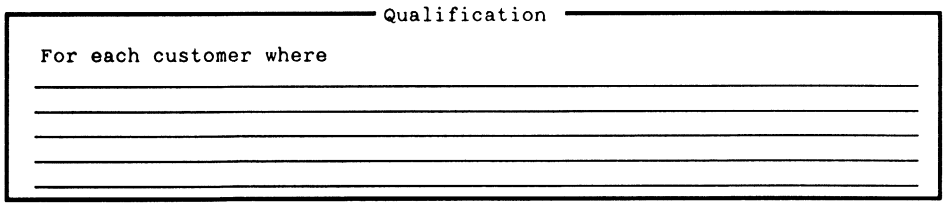


Figure 6–12: The Free Form Qualification Window

Chapter 7
Maintenance

This chapter devotes a section to each Maintenance menu option. Many sections also contain subsections explaining additional important information.

This chapter describes how the Maintenance menu provides the following database services:

- Deleting a FAST TRACK object from the database.
- Dumping and loading FAST TRACK data files.
- Distributing your FAST TRACK application.
- Compiling your application.
- Using Run-time security to restrict access.
- Targeting printer/output destinations.
- Generating Development Reports.
- Changing database names in you FAST TRACK database.

FAST TRACK provides several utilities to enhance PROGRESS application program maintenance and development. When you choose the `Maintenance` option from the Main Menu, FAST TRACK displays the menu shown in Figure 7-1.

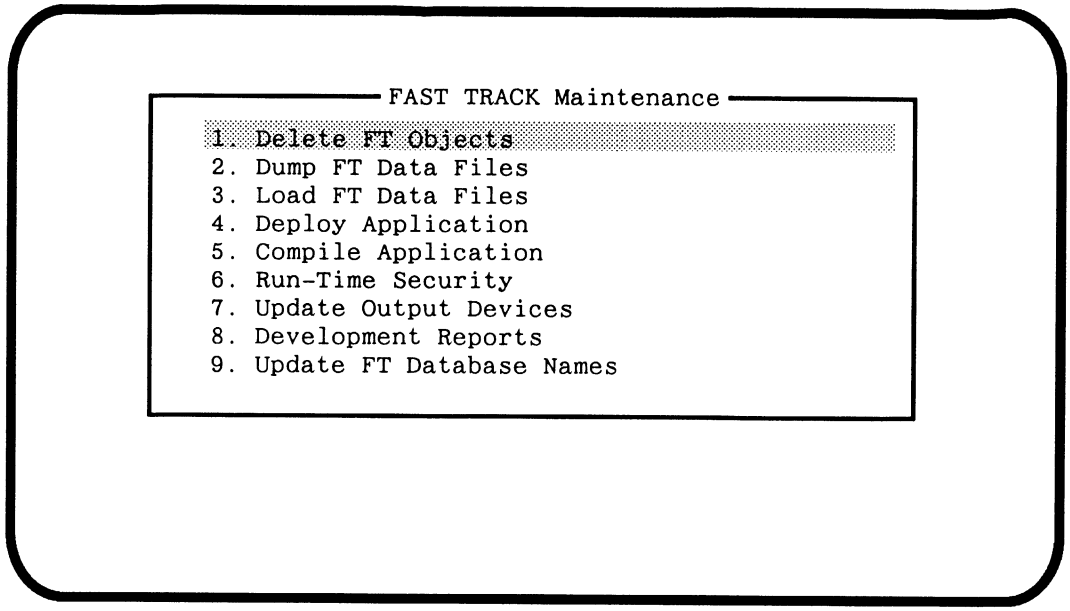


Figure 7-1: The FAST TRACK Maintenance Menu

Select options from the Maintenance menu in the same manner as you select options from the Main Menu — namely, by typing the associated number; typing the first set of unique letters corresponding to a menu option; or moving the selection bar with the cursor keys and pressing [RETURN]. Refer to Chapter 2 for additional information on using vertical menus.

7.1 DELETING FAST TRACK OBJECTS

The `Delete FT Objects` option deletes FAST TRACK objects from your database. An object is a series of records in the FAST TRACK database that define a given FAST TRACK component such as a QBF or menu.

If you use the `Delete FT Objects` option, all references to the object are deleted from the FAST TRACK database. This includes all objects related to the deleted object, unless the deleted object is a QBF. Additionally, any generated files associated with the object are deleted.

To delete a menu, report, form, or QBF object from your FAST TRACK database, follow these steps:

1. Choose `Delete FT Objects` from the FAST TRACK Maintenance menu. FAST TRACK displays the warning shown in the following figure.

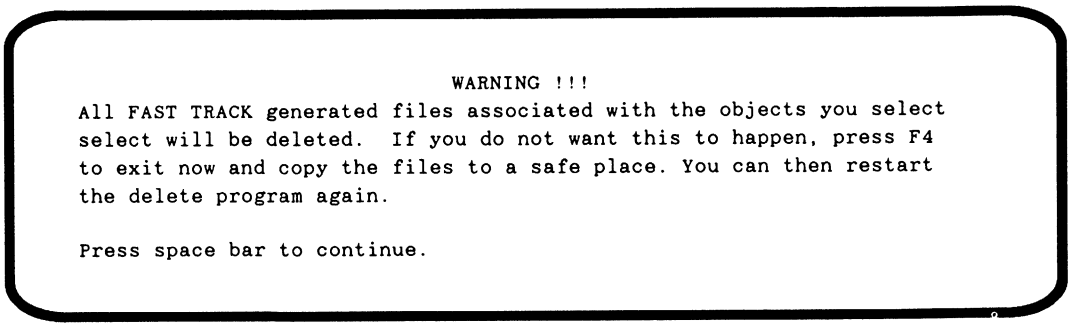


Figure 7-2: Warning for Delete FT Objects Screen

2. If you want to delete all FAST TRACK generated files associated with the objects you select in the next step, press SPACEBAR. FAST TRACK displays the window shown in the following figure. The window prompts you for the type of FAST TRACK object to delete.

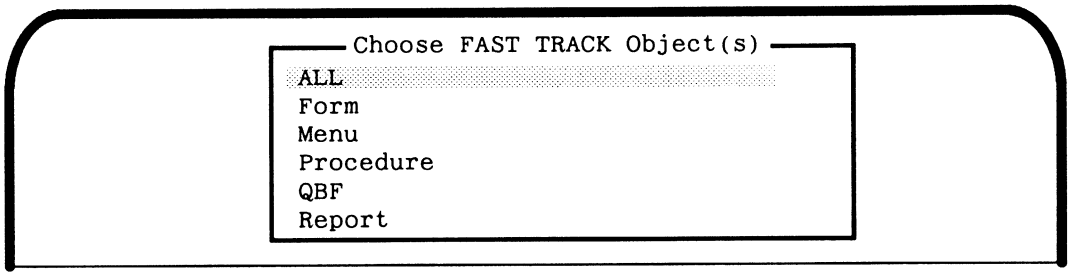


Figure 7-3: FAST TRACK Objects

When you make a selection from this window, FAST TRACK automatically displays the Choices window with all the objects of the selected type in your database. For example, when you select `Report`, FAST TRACK displays a list of all your report objects as illustrated in the following figure.

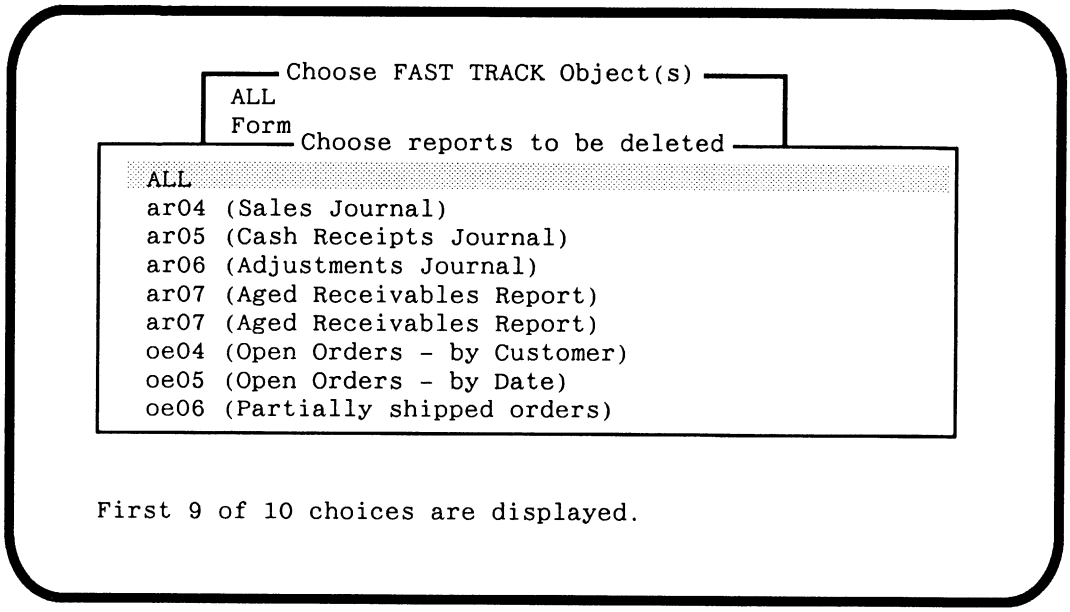


Figure 7-4: FAST TRACK Objects

3. Mark a choice to delete by highlighting it with ↑ and ↓, and pressing RETURN. An asterisk (*) appears beside the choice to indicate that it has been chosen. If you make a mistake, you can unmark a choice by highlighting it again and pressing RETURN. When you have marked all of the items that you want to delete, press GO (F1).

4. FAST TRACK displays confirmation messages as it deletes marked objects. When it finishes the deletion process, it prompts you to press the SPACEBAR to return to the Objects menu.

5. To delete other types of FAST TRACK objects, select the object type by highlighting it, and repeat steps 3 and 4.

7.2 DUMPING AND LOADING DATA FILES

The procedure for dumping and loading a FAST TRACK database is important to both the application development process and program maintenance.

During application development — or while using a FAST TRACK application database — a number of records could have been added to the database and subsequently deleted. In the course of deleting records, space in the database becomes available for the addition of new records. This space can be made available to the system — thereby reducing the size of the database — by using the dump and reload options.

You can also use the dump and load options to create an empty start-up version of your application database.

NOTE: You can use the dump and load procedures on FAST TRACK databases only. To convert a PROGRESS database to a FAST TRACK database, you must use the `convft` utility, which is explained in Chapter 1.

For more information about dumping and loading PROGRESS database files, see Chapter 4 of the *System Administration II: General* manual.

7.2.1 Dumping FAST TRACK Data Files

When you invoke the `Dump FT Data Files` option from the FAST TRACK Maintenance menu, FAST TRACK prompts you to confirm whether you want to dump the data files for the current database. Answer `y` to proceed.

As FAST TRACK performs the dump process, it displays a message for each file it dumps. Note that FAST TRACK appends a `.d` extension to each data file it dumps. When the dump process concludes, press the SPACEBAR to return to the Maintenance menu.

For additional information on data files, refer to Appendix D.

7.2.2 Loading FAST TRACK Data Files

Before you load FAST TRACK data files, ensure that you are working with a newly created FAST TRACK database. Also ensure that all the FAST TRACK data files (`.d`) that you want to load are in your application directory.

You can create a new FAST TRACK database from an existing PROGRESS database by using the `convft` command. You can also create a new empty FAST TRACK database using the `prodb` command. Refer to Chapter 1 for additional information on using `convft` and `prodb`.

Once you have a new FAST TRACK database, select the `Load FT Data Files` option from the FAST TRACK Maintenance menu. FAST TRACK loads all of the data files previously dumped. As files are loaded, FAST TRACK displays a series of status messages. When the load process concludes, press the SPACEBAR to return to the Maintenance menu.

7.3 DEPLOYING YOUR APPLICATION

After you create a FAST TRACK application, you can consolidate the application files for the operating system under which it was created, or distribute it for a different operating system. Both of these tasks can be accomplished by using the `Deploy Application` option on the Maintenance menu.

In order to run a deployed application on another computer system (even one with the same operating system under which the application was created) the end-user must have one of the following versions of FAST TRACK:

- FAST TRACK
- PROGRESS Query/Report
- FAST TRACK Run-time Utilities

To realize the full functionality of a FAST TRACK application, you must use a FAST TRACK database and run it with PROGRESS 4GL/RDBMS or PROGRESS Query/Run-time.

A FAST TRACK database is supported by FAST TRACK drivers, no matter what version of FAST TRACK is installed. If you use PROGRESS Run-time, however, you do not have the capability to compile procedures. Accordingly, you should ensure all your code is compiled before installation. Here are some guidelines for developing FAST TRACK applications for PROGRESS Run-time:

- QBF records cannot be qualified at run time.
- Include files associated with output devices cannot be modified, either with the Report Writer's Ask option, or via the `Update Output Devices` option on the Maintenance menu.
- A report's page size cannot be changed by the end-user, either with the Report Writer's Ask option, or via the `Update Output Devices` option on the Maintenance menu.

You can run FAST TRACK reports on any PROGRESS database. The report code, however, cannot access FAST TRACK's run-time security and output device control options, with the exception of the `Ask` device control option. To use the `Ask` option, you must ensure that the following include file is invoked by the application's main menu or by a custom start-up procedure:

```
{ft/agshrr.i new global}
```

When you deploy an application to a different operating system, you should be aware of differences between the source and target operating systems. This is an especially important consideration when naming files. For example, DOS uses filenames of eight characters (with a three character extension), while UNIX, BTOS/CTOS, and VMS use much longer filenames.

7.3.1 Dumping Your Application

The first step in deploying your FAST TRACK application is to dump your application from the source machine.

When you undertake the deployment dump process, do so with the target system in mind. For example, if you have developed an accounting system consisting of general ledger, accounts payable and accounts receivable modules, you can deploy the whole system or a subsystem of it.

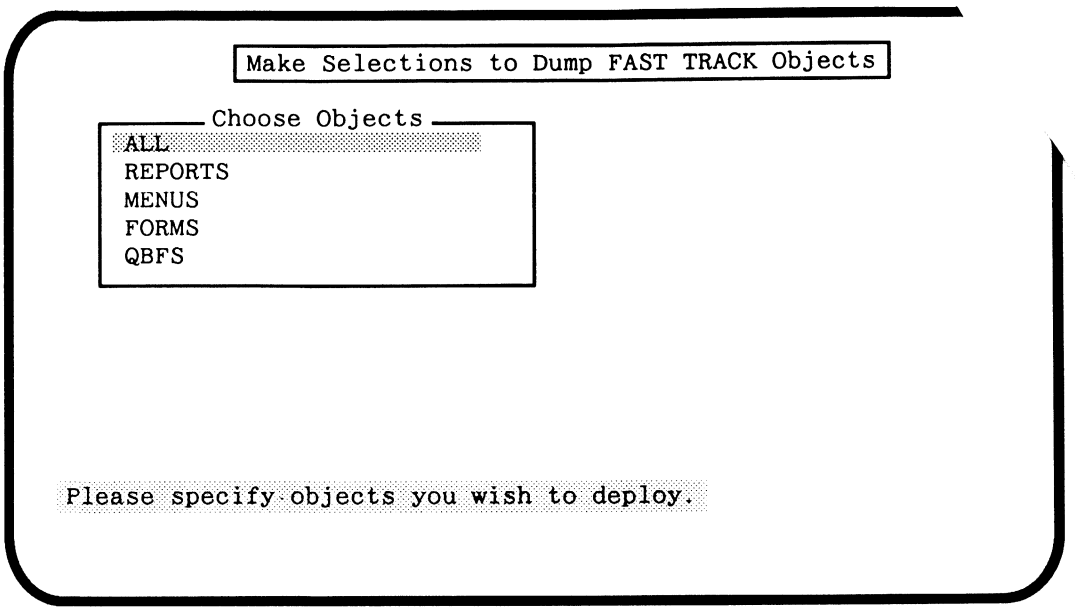


Figure 7-7: Objects to Dump

You can choose to dump all objects or a subset. The following list explains your choices:

ALL	Dumps reports, menus, screens, and QBFs.
REPORTS	Dumps reports created using FAST TRACK.
MENUS	Dumps all menus and any associated objects.
FORMS	Dumps forms created using FAST TRACK.
QBFS	Dumps QBFs created using FAST TRACK.

4. After you have made your selection(s) by highlighting each type of object and pressing [RETURN], press [GO] (F1). If you select ALL, FAST TRACK dumps the objects in the following order: reports, menus, screens and QBFs. Otherwise, it dumps them in the order of your selection.

If you select REPORTS, MENUS, FORMS, or QBFS, FAST TRACK displays a second window th gives you the choice of dumping all objects of the same type, or a specified subset of all objects o same type.

The following sections describe the dump process for each type of FAST TRACK obje

MENUS. If you want to dump menus, select this option from the Choose Objects v TRACK then displays your menu objects in the Choose Menus window. Figure 7-' example.

If each subsystem has its own top-level menu, you can dump the top-level menu using the option to dump all its associated objects. This avoids dumping objects that are not part of the subsystem. Alternatively, you can dump all objects in your database.

NOTE: Store your dumped reports, menus, screens and QBFs in different subdirectories, because the report and QBF dump procedures create files with the same names.

The following steps describe how to dump your application files:

1. Choose the **Deploy Application** option from the FAST TRACK Maintenance menu. FAST TRACK displays the window shown in Figure 7-5 .

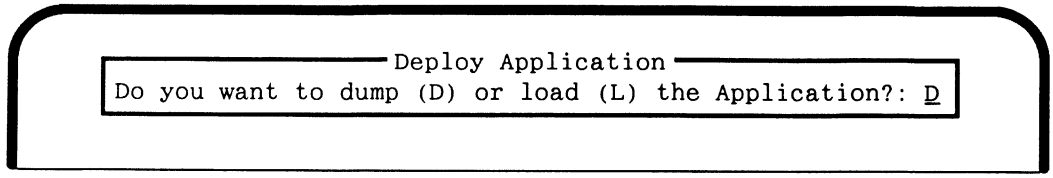


Figure 7-5: Deploy Application Window

2. Type **D** to dump the application. FAST TRACK displays the following information screen on the deploy process.

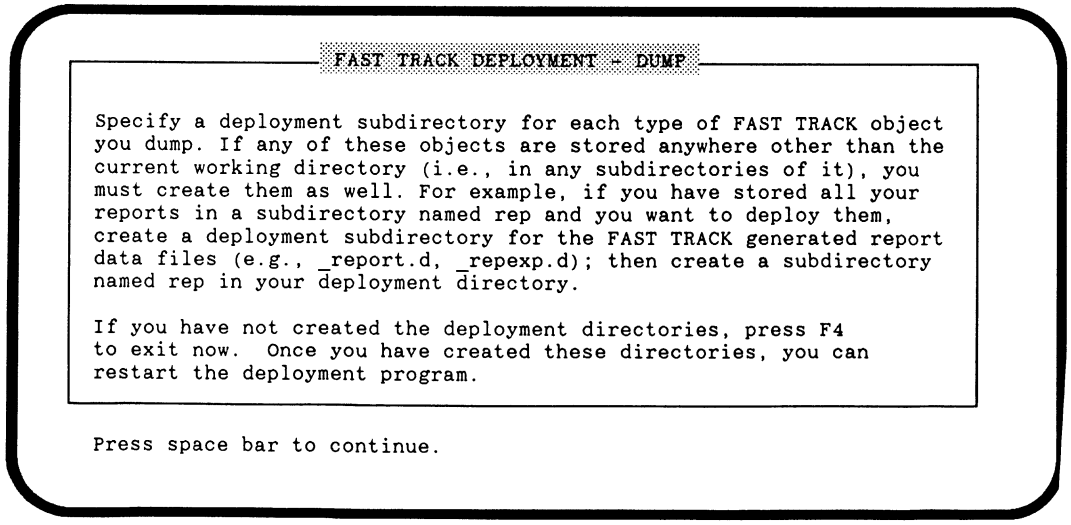


Figure 7-6: Deployment Dump Window

3. Press the SPACEBAR to continue.

   FAST TRACK displays the Choose Objects window and prompts you to select the obje that you want to dump. Figure 7-7 shows the Choose Objects window.

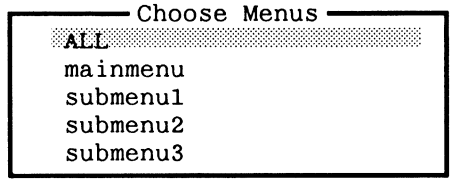


Figure 7-8: Menu Objects to Dump

After you make your selection(s) by highlighting each type of object and pressing RETURN, press GO (F1). At this point, FAST TRACK prompts you for a deployment subdirectory. The default for menu objects is menudep (which stands for menu deployment). You can enter any directory name you want, however. If you do not want your deployment directory created in your current directory, enter the full pathname of the deployment directory.

If a directory already exists, FAST TRACK displays an error messages, but continues deployment by overwriting any files that have the same name as the files being deployed.

NOTE: On UNIX and VMS systems, ensure that you have adequate permission to create files and write files to existing directories. If, during the deployment process, you discover you do not have adequate permission, press CTRL-C to cancel deployment.

FAST TRACK prompts you to specify whether you want to dump a compiled (.r) file with each procedure (.p) file. If you deploy to a target system that is the same as the source system, deploy the compiled code. If you deploy to a target system that is different from the source system, deploy only the menu files.

During the deployment process, FAST TRACK confirms the type of files it is deploying. When FAST TRACK finishes deployment, it displays the following message:

```
Deployment dump program complete.
Press space bar to continue.
```

Your deployment directory now contains your menu files and any associated objects tied to a top-level menu. In addition, the deployment directory contains the following data files:

```
_menu.d
_choice.d
_menprog.d
```

Refer to Appendix D for additional information on data files.

REPORTS. To dump FAST TRACK reports, select this option from the Choose Objects window. FAST TRACK then displays your report objects in the Choose Objects window. The following figure shows a typical example.

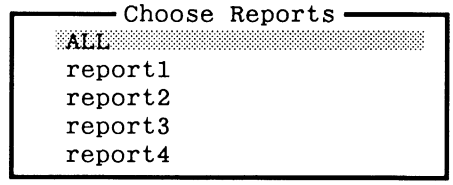


Figure 7-9: Report Objects to Dump

To dump reports, follow the same steps outlined for dumping menus. The major difference is that the default deployment directory is `repdep`. As with deploying menus, you can specify a different deployment directory.

The deployment process copies all report related files to the deployment directory. These consist of procedure (`.p`) and include (`.i`) files. If you choose to deploy compiled (`.r`) files, these files are also deployed. In addition, the following report-related data files are copied to the deployment directory:

```
_fdef.d
_form.d
_ragg.d
_repexp.d
_report.d
_rgroup.d
_rlevel.d
_row.d
_rqual.d
_repprog.d
```

Refer to Appendix D for additional information on data files.

QBFS. If you want to dump QBFs, select this option from the Choose Objects window. FAST TRACK then displays your QBF objects in the Choose QBFs window. Figure 7-10 shows a typical example.

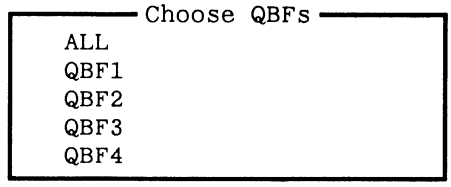


Figure 7-10: QBF Objects to Dump

Dumping QBFs follows the same steps as dumping menus. The default deployment directory is `qbfdep`. As with deploying menus, you can specify a different deployment directory. But remember that default QBF procedures are put in a subdirectory named after the database name, unless you specified other subdirectory name for the generated procedures from the OBF Generation Options Menu.

If you have default QBF procedures stored in subdirectories (either by default or user specified), then before attempting deployment, create the subdirectories under `qbfdep` (or the subdirectory specified) to make sure the subdirectory structure is there in the target deployment directory.

NOTE: In addition to dumping QBF objects, QBF deployment dumps all screen objects used to generate the QBF. If all your screens are associated with a QBF, you do not need to dump the screens separately using the Screen deployment option.

The deployment process copies all QBF related files (and any files for QBF screens) to the deployment directory. These consist of procedure (`.p`) and include (`.i`) files. If you chose to deploy compiled (`.r`) files, these are also deployed. In addition, the following QBF related data files are copied to the deployment directory:

```
_form.d
_fdef.d
```

Refer to Appendix D in this manual and the *Programming Handbook* for additional information on data files.

FORMS. If you have not already dumped all your forms using the QBF deployment option, select this option. FAST TRACK then displays your report objects in the Choose Forms window. Figure 7–11 shows a typical example.

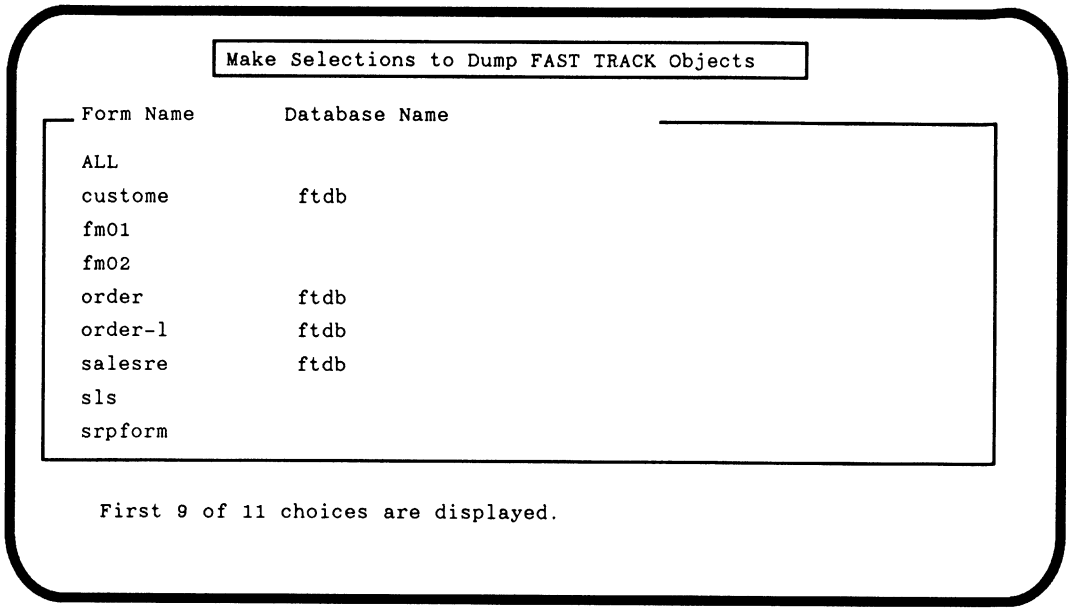


Figure 7-11: Form Objects to Dump

Dumping forms follows the same steps as dumping menus. The major difference is that the default deployment directory is `formdep`. As with deploying menus, you can specify a different deployment directory.

The deployment process copies all FAST TRACK form procedure files to the deployment directory. These consist of form (`.f`), and compiled (`.r`) files. In addition, the following form related data files are copied to the deployment directory:

```
_fdef.d
_form.d
```

Refer to Appendix D for additional information on data files.

7.3.2 Installing Your Application

Before attempting to install your application, ensure that the environment on the target system is set up properly.

The FAST TRACK installation script sets the DLCFT environment variable to point to the FAST TRACK system directory. If you have moved the FAST TRACK software to a directory other than the one in which you originally installed it, reset DLCFT. Refer to the *Programming Handbook* for additional information on environment variables.

If you have dumped your application in separate subdirectories for menus, QBFs, reports, and PROGRESS procedures, maintain the same subdirectory structure on the target system. Once you have set up the environment of the target system, copy the deployment dump files from the delivery media to the target system.

NOTE: When deploying an application, several factors can affect the deployment. You must ensure that the target FAST TRACK product is the product for which you developed your application. Additionally, consider whether the deployment involves a new application installation, or whether it involves upgrading an existing PROGRESS application.

The following sections describe how to deploy files for the various FAST TRACK products.

Deploying to FAST TRACK and PROGRESS Query/Report. The deployment process for FAST TRACK and PROGRESS Query/Report is generally the same. The following steps explain how to deploy an application with either of these FAST TRACK products:

1. If the target system is a new installation, create an empty FAST TRACK database using `prodb`. The FAST TRACK empty database is named `emptyft`, and is located in the FAST TRACK system directory. Refer to Chapter 1 for information on creating a FAST TRACK database.

2. If you want to deploy FAST TRACK application code to an existing PROGRESS application, run the `convft` utility on that database to convert it to a FAST TRACK database. Ensure that you back up the database before you run `convft`. Refer to Chapter 1 for details on how to run `convft`.

Now you are ready to deploy your application. Note that if you have a FAST TRACK database, you can start FAST TRACK. See Chapter 1 for details on how to start a FAST TRACK database.

Deploying To FAST TRACK Run-time Environments. The FAST TRACK Run-time Utilities are distributed as part of PROGRESS 4GL/RDBMS, PROGRESS Query/Run-time, and PROGRESS Run-time products.

The following steps explain how to deploy an application to PROGRESS environment that contains the FAST TRACK Run-time Utilities:

1. If the deployment involves installing a new application, create an empty FAST TRACK database from `emptyft` (see Chapter 1).

2. Run the FAST TRACK utility `ftload` on your existing database or the empty FAST TRACK database that you create. The `ftload` utility is described in the next section. Once `ftload` finishes, FAST TRACK returns to the Maintenance Menu.

When you want to deploy a FAST TRACK application and compile it on a target machine using PROGRESS Run-time or Query/Report products, you need to perform the following tasks:

1. Encrypt all FAST TRACK generated source code and all FAST TRACK Run-time source code (menu- an qbf-drivers).
2. If you don't know the names of all run-time procedures to encrypt, you can encrypt all FAST TRACK source code (on files with `.p` and `.i` extensions in all FAST TRACK system subdirectories) on your full system, and distribute that together with the encrypted application. Of yourse, you must keep the directory structure when you install on the target machine.
3. When you compile, use `_proxcom`. Do not use _progres. If you use `ftload-script`, just change all `_progres` to `_proxcom`.

The ftload Utility. The `ftload` utility is designed to convert a PROGRESS (version 4.2E or later) database to a FAST TRACK database. The `ftload` software is packaged with FAST TRACK and installed in the main FAST TRACK directory. Additionally, `ftload` loads the Maintenance menu, which is your entry point into FAST TRACK when you deploy an application.

To run `ftload` on you system, use the appropriate command:

Table 7-1: ftload Command

Operating System	FAST TRACK ftload Command
UNIX	`ftload` *db-name*
DOS	`ftload` *db-name*
VMS	`@FTLOAD` *db-name*
BTOS/CTOS	`FAST TRACK Convert Database` `Database Name` *db-name*

Use the actual name of your database instead of *mydemo*. If the database is already a FAST TRACK database, `ftload` displays a message to inform you of this.

If the database is not a FAST TRACK database, FAST TRACK displays the following message:

```
This is not a FAST TRACK database.
Do you wish to convert it(y/n)?
```

If for any reason you do not want to proceed, type **n** and the program terminates. Otherwise, type **y**.

If your database needs conversion, you are asked for confirmation that you have done a backup of your system. Type **y** to confirm, or **n** to exit. When `ftload` finishes converting your database, it displays the following message:

```
Successful FAST TRACK conversion completed.
```

In a short time, `ftload` executes and displays the FAST TRACK Maintenance Menu, from which you can load your application by following these steps:

1. Choose the `Deploy Application` option from the FAST TRACK Maintenance menu. FAST TRACK displays the window shown in Figure 7–12.

```
Deploy Application
Do you want to dump (D) or load (L) the Application?: D
```

Figure 7–12: Deploy Application Window

2. Type **L** to load the application (note that the default is `D`). FAST TRACK displays the following information screen about the deploy process.

```
FAST TRACK DEPLOYMENT - LOAD

For each application you deploy, specify an application directory. This
can be either your current working directory or any directory you name.
In addition, for each type of FAST TRACK object you load, specify a
deployment source directory. FAST TRACK reads data files (e.g., _qbf.d,
_repord.d) from these source directories. It also copies your appli-
cation programs from the source directories to your application
directory. If your application programs are structured in subdirectories,
create the same directory structure in your application directory.
For example, if your report deployment directory is named repdep and
your reports are in a subdirectory named rep in repdep, you must create
a subdirectory named rep in your application directory.

WARNING: If you are loading an object with the same name as an existing
         object, the object in your database will be overwritten !

If you need to create any subdirectories or want to stop the load,
press F4 to exit now.

Press space bar to continue.
```

Figure 7–13: Deployment Load Window

3. Press the SPACEBAR to continue.

4. FAST TRACK displays Choose Objects window and prompts you to select the objects that you want to load. Figure 7–14 shows the Choose Objects window.

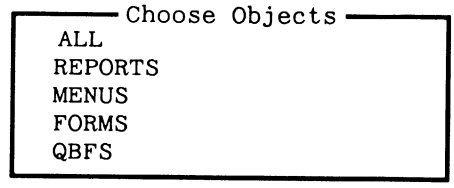


Figure 7-14: Choose Objects Window

5. You can choose to load all objects or a subset. The following list explains your choices:

ALL	Loads reports, menus, screens, and QBFs.
REPORTS	Loads reports created with FAST TRACK.
MENUS	Loads menus created with FAST TRACK.
FORMS	Loads all form objects for an application.
QBFS	Loads all QBF objects for an application.

After you make your selection(s) by highlighting each type of object and pressing [RETURN], press [GO] (F1). If you selected ALL, FAST TRACK loads the objects in the following order: reports, menus, forms and QBFs. Otherwise, it loads them in the order of your selection.

FAST TRACK prompts you for a deployment subdirectory name for each object type. If you have dumped all your objects into one directory, enter the the name of this directory in response to each FAST TRACK prompt. For example, if you dumped a subsystem of your application into a single directory by dumping its main menu (and thus, its associated submenus, reports and QBFs), you must load your reports, menus, forms, and QBFs from this same directory.

To load an object, select the object from the Choose Objects window. FAST TRACK responds asking you to specify a deployment subdirectory. For example, if you selected MENUS from the Choose Objects window, FAST TRACK responds with the following prompt:

```
Specify deployment subdirectory for menus:
menudep
Application Directory:
```

If you do not specify a deployment directory, FAST TRACK uses the default deployment directory. In the previous example, the default deployment directory is menudep. Table 7-2 lists all the default deployment directories.

Table 7–2: Default Deployment Directories

Object	Default Deployment Directory
Menus	menudep
Reports	repdep
QBF's	qbfdep
Screens	formdep

When you use the deployment load option, take into account the following considerations:

- Different users can run the database from different directories, so the application directory can be different from the current directory.
- In addition to loading QBF objects, the QBF load option loads all screen objects used to generate the QBF. If all your screens are associated with a QBF, you do not need to load the screens separately using the Screen load option.
- If you inadvertently use the Screen load option after having used the QBF load option to load your screens, FAST TRACK displays warning messages about duplicate forms. If this happens, allow FAST TRACK to complete the Screen load operation; it simply overwrites the previous copies of your screens.

When FAST TRACK finishes loading your application objects, it displays the following message:

```
Deployment load program complete.
Press space bar to continue.
```

You can now access your application objects on the target system. To do so, execute the application as you would have your user execute it. Be careful if you decide to upgrade to a FAST TRACK system after running an application in a FAST TRACK Run-time environment: if you install an upgraded application in the same directory, your old application is overwritten. If you do not want this to happen, install the upgraded application in a different directory.

NOTE: If you intend to deliver a FAST TRACK application to run on hardware and/or operating systems that are different from your development system, or if you intend to compile your application system at a site without PROGRESS 4GL/RDBMS installed, you need the PROGRESS Developer's ToolKit to compile your code.

7.4 COMPILING YOUR APPLICATION

You need to compile an application when you deploy it to a target machine with a different operating system from the one on which you developed the application. Sometimes it is also necessary to compile an application after you have made a change to the database schema. In both these cases, use the `Compile Application` option on the Maintenance menu.

If you want to generate QBFs for several files when running on a DOS system, you should generate the QBFs without compiling or running them in order to conserve memory. Instead, generate them without the compile and run options and compile them later using the `Compile Application` option of the Maintenance Menu.

The `Compile Applications` option lets you compile FAST TRACK applications and/or PROGRESS procedures that are invoked by FAST TRACK menus. Note that the `Compile Applications` option does not compile other types of PROGRESS procedures.

7.4.1 Prompts and Messages

When you select the `Compile Application` option, FAST TRACK prompts you to confirm whether you want to proceed with the compilation. The reason for this is to avoid inadvertent use of the `Compile Application` option. If you confirm by typing **y**, FAST TRACK displays each item it compiles in the message area:

```
Compiling <program1>.p
Compiling <program2>.p . . .
```

After compiling a file, FAST TRACK pauses until you press the SPACEBAR to proceed to the next file. If you do not press the SPACEBAR within five seconds, FAST TRACK proceeds anyway. When FAST TRACK finishes compiling all files, it displays a completion message.

7.5 RUN-TIME SECURITY

The `Run-Time Security` option on the FAST TRACK Maintenance menu allows you to define access restrictions for the various components (menus, QBFs, reports, screens) that you have integrated into your application. When you select this item from the Maintenance menu, the menu shown in the following figure appears.

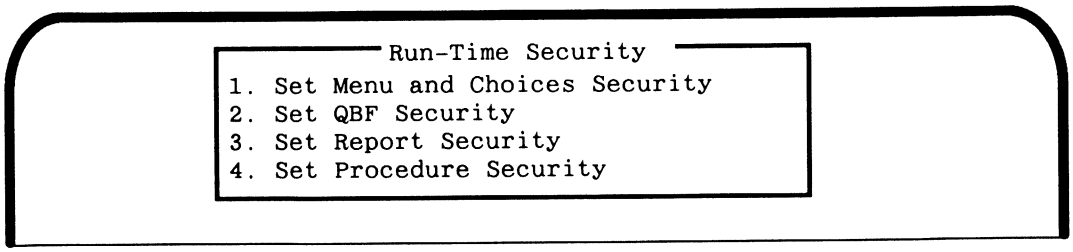


Figure 7-15: The Run-Time Security Menu

NOTE: In order to use Run-Time Security, you have to run on a FAST TRACK database. If you want to run a QBF or Report on a non-FAST TRACK database, then you have to generate the QBF or Report with a question mark "?" at the `can be run by` option in the QBF or `Report Writer` menu. But note that QBFs and Reports generated with a "?" will not show up in the `Security Selection Menu` shown in Figure 7-16, because you cannot change their security status. If you wan to add security to a QBF or Report generated without the "?", you have to go back to the Report Writer or QBF Generator and regenerate with a new `can be run by` value.

When you select the `Set Menu and Choices Security` item from this menu, the Display/Change Runtime Menu & Choice Access window appears as shown in Figure 7-16.

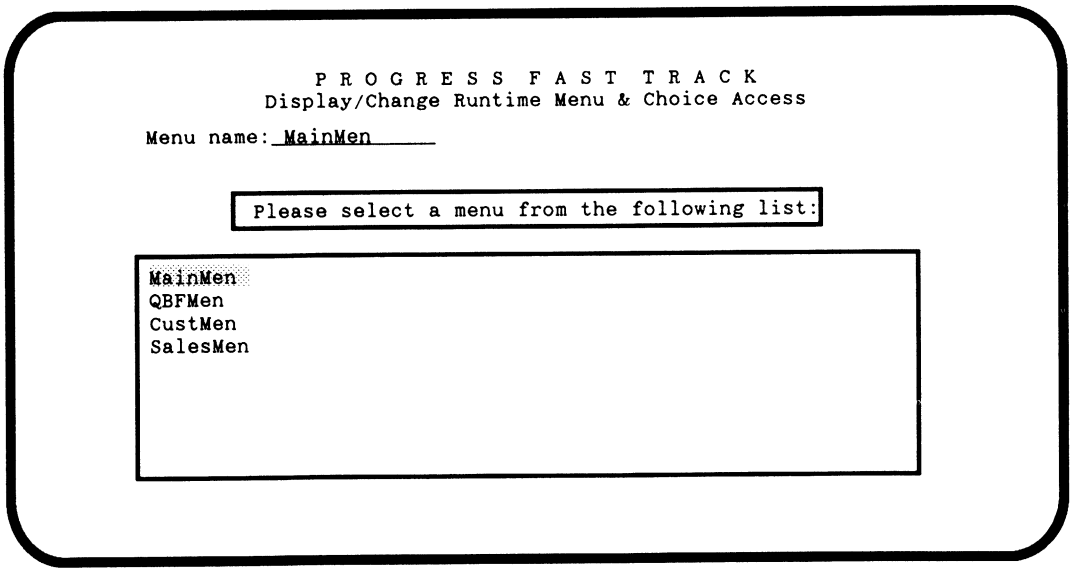


Figure 7-16: Security Selection Menu

After you select a menu, the window shown in the following figure appears.

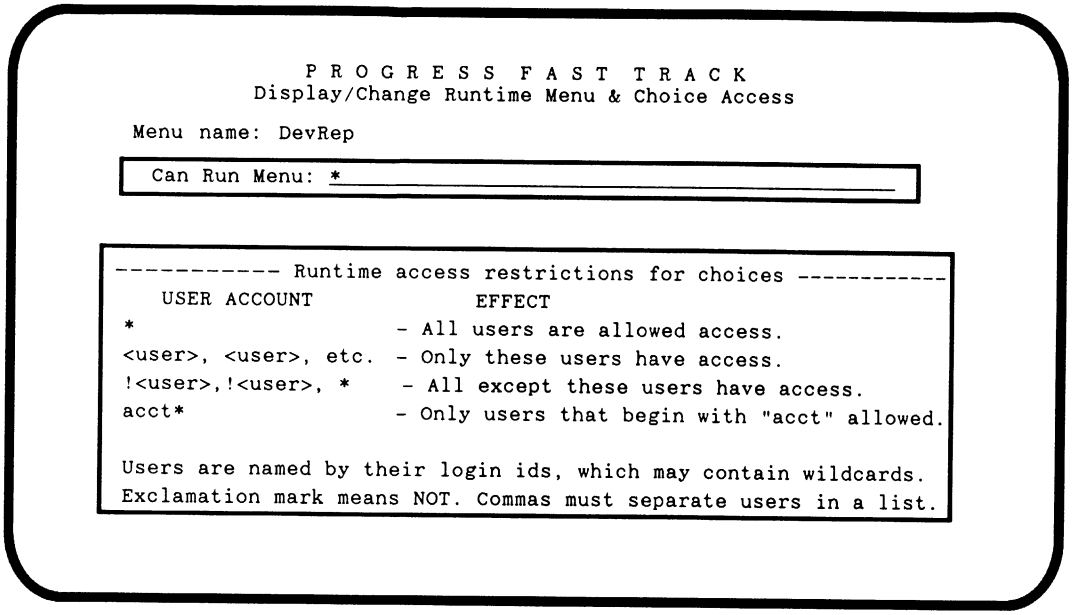


Figure 7–17: Menu and Choice Access Window

A help window provides information on how to complete the Can Run description. What you enter in the Can Run window determines who can run the particular menu or menu choice. The possibilities are as follows:

- Asterisk. Denoted by *, this is the default setting, and indicates that all users have access to the menu. The asterisk is a wildcard character used in PROGRESS.
- A List of Users. A list of user names, separated by commas, indicates that *only* the specified users have access to the menu as in the following example:

```
nancy,tom,rick,chip
```

- A List of Excluded Users . A list of names, each preceded by and exclamation point (!), and followed by an asterisk (*), indicates that all users except those specified have menu access. The following example shows the correct syntax.

```
!nancy,!tom, *
```

- Any Sequence of Letters Followed by *. A sequence of letters followed immediately by an asterisk (*) indicates that only those users whose names begin with that sequence of characters have menu access. The following example shows the correct syntax:

```
acct*
```

After you set access privileges for the menu, the Choice Security window appears as shown in the following figure.

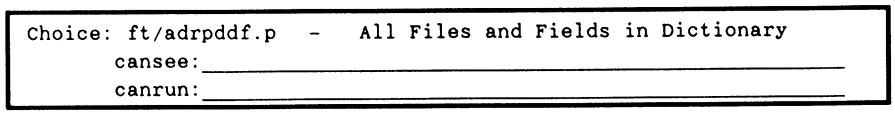


Figure 7–18: Choice Security Window

The top line in this window gives the action associated with the choice, and the text of the menu option. In the window, you can specify two additional access privileges.

- `cansee`. Specifies who is allowed to view the particular menu item.
- `canrun`. Specifies who can run the menu item.

The access privileges for these two items follow exactly the same format as for `Can Run Menu` explained earlier in this chapter.

For reports, QBFs, and procedures, you specify access privileges in the same way as for menus.

See the *Programming Handbook* for additional information on PROGRESS security.

7.6 UPDATE OUTPUT DEVICES

The `Update Output Devices` option on the Maintenance menu lets you modify the output device specifications for a report. This option is designed for system administration and/or the developer making an on-site installation.

Initially, you define output devices during the process of creating a report in FAST TRACK's Report Writer module. The `Update Output Devices` option is appropriate to use when you need to give a single user access to more than one printer — or when different users require access to unique disk files, include files, or spoolers.

NOTE: In order to use the `Update Output Devices` option, you must define a report's output classes using the Report Writer. You cannot define new classes using the `Update Output Devices` option.

7.6.1 Update Output Classes

When you invoke the `Update Output Devices` option, FAST TRACK displays the Update Output Classes window shown in the following figure.

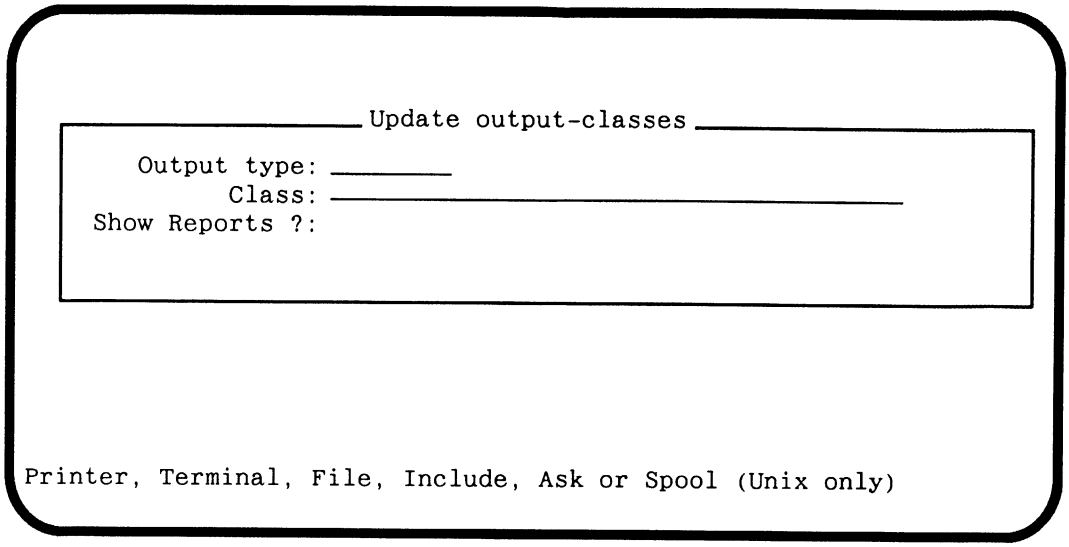


Figure 7-19: Update Output Classes Window

The first two fields in this window — the `Output type` and `Class` fields — are critical to correctly setting up output devices. For each type and class defined in the Report Writer, you can define one or more output devices, which can later be assigned to different users. This window is where you specify the type and class as it appears in an existing report.

The following sections describe the correct responses for the Update Output Classes window. Press CHOICES (ESC C) to display the Choices window with a list of correct responses.

Output type. In the `Output type` field, you must enter one of the four output categories that can be specified in the Update Output Devices menu. The following comprise valid `Output type` responses:

- *Printer* - any standard printing device such as an 80 or 132 column line printer or a laser printer.
- *Spool* - on UNIX systems only, any program through which you want to filter your output.
- *File* - an ASCII file on the specified media.
- *Include* - a PROGRESS include file that contains code specifying the output device.

Class. In the `Class` field, enter a symbolic name representing the type of output device. For example, if the output device is an 80-column line printer, **prn80** is a typical `Class` name.

Show Reports. In the `Show Reports` field, enter **yes** or **no** to indicate whether you want to list reports currently assigned to an output class. If you indicate **yes**, FAST TRACK displays the window shown in the following figure.

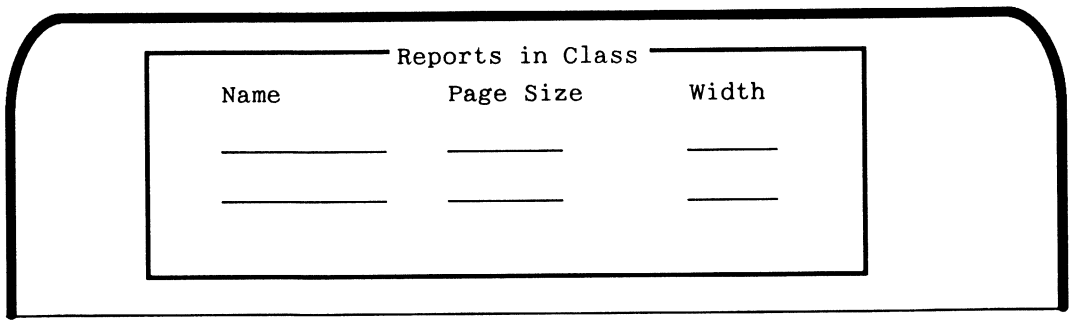


Figure 7–20: Reports in Class Window

7.6.2 Add New Devices

When you specify a new device in the Update Output Classes window, press GO (F1) and FAST TRACK displays the Add New Devices window shown in Figure 7–21.

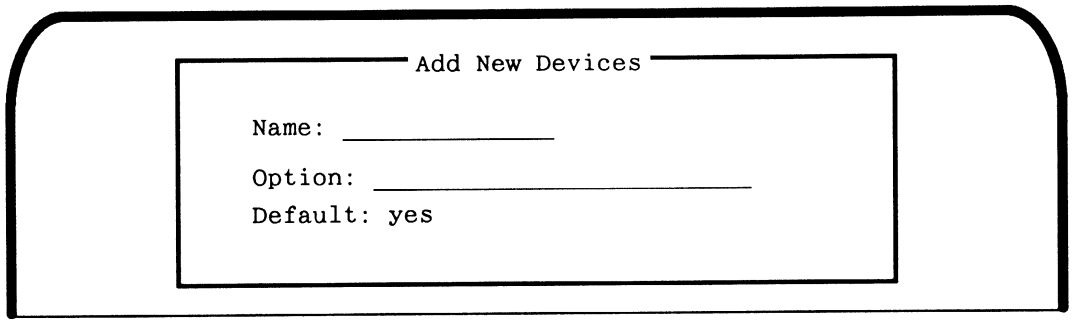


Figure 7–21: Add New Devices Window

The following sections describe the fields in the Add New Devices window. Press GO (F1) when you have finished entering information into the Add New Devices window.

Name. In the Name field, enter a unique name for the device. It is a good idea to choose a name that corresponds to a logical designation for the device. For example, if you had two line printers and one laser printer, call them *lpr1*, *lpr2*, and *lprn*.

Option. In the Option field, enter the logical parameters based on your operating system. Table 7–3 lists typical parameters.

Table 7–3: Output Device Options

Device Type	OS	PROGRESS Statement	Description
Printer	UNIX	`OUTPUT THROUGH` *option*	Printer spool program
	DOS	`OUTPUT TO` *option*	Printer device name
	VMS	`VMS DEFINE/JOB SYS$PRINT` *option* `OUTPUT TO PRINTER`	Output queue name
File	All	`OUTPUT TO` *option*	Name of disk file
Include	All	{*option*}	Name of include file
Spool	UNIX	`OUTPUT THROUGH` *option*	Name of UNIX program

Default. In the `Default` field, enter **yes** or **no** depending on whether you want to make the current device the default device. You can only specify one default device per output class (see the section titled "Adding and Unassigning Users").

7.6.3 Update Devices

FAST TRACK prompts you to update a device, if you choose not to add a new device to the system. When you choose to update a device, FAST TRACK displays the Update Device window, which gives you the opportunity to change the `Option` and `Default` fields.

The `Update Output Devices` option also lets you delete devices. The Update Devices window displays a list of devices for you to update. Additionally, it gives you the choice of deleting all devices. You mark a device for deletion with the RETURN key. After you have finished the marking process, press GO (F1) and FAST TRACK deletes the devices.

7.6.4 Adding and Unassigning Users

After you have entered an output device, press GO (F1) and FAST TRACK displays the Add New Users window. In this window, you can assign a device to one or more users.

To assign a user to a device, enter the User Id in the Add New User window. This is accomplished by simply typing the user's id into the window. Figure 7–22 shows the Add New Users window with an example user id.

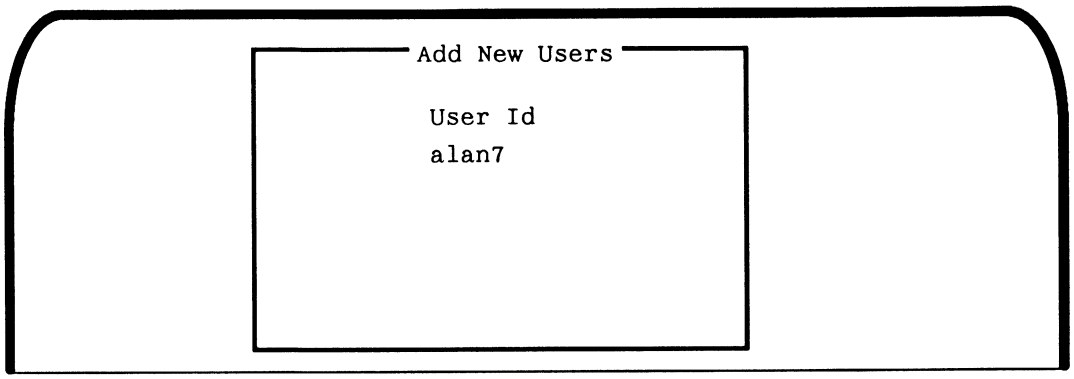


Figure 7-22: Add New Users Window

When FAST TRACK runs a report, it checks if the user has any assigned device for the actual type and class. If it cannot find an assigned device, FAST TRACK designates the default device for the type and class.

To remove a user from a device, use the Unassign Users window. This window automatically appears after you complete the Update Devices window. The Unassign Users window displays a list of users for you to remove from a device. Additionally, it gives you the choice of unassigning all users. You mark a user for removal with the RETURN key. After you have finished the marking process, press GO (F1) and FAST TRACK unassigns the user.

7.7 DEVELOPMENT REPORTS

During the development of an application, the number of fields, forms, QBFs, reports, and menus can proliferate. It can become very difficult to keep track of the organization of the application and the definition of its components. FAST TRACK provides you with a rather extensive set of reports that allow you to monitor all aspects of the development and definition of your application.

In the Menu Editor, Screen Painter and Report Writer, when you press OPTIONS (CTRL-O) and select OTHER→REPORTS, FAST TRACK allows you to view development reports, providing you with the menu that appears in Figure 7-23.

7.7.3 Dictionary Information on One File

When you choose the `Dictionary Information on One File` option on the Development Reports menu, FAST TRACK prompts you for the name of a file in the current database and then displays the fields that are defined for that particular file. This is the information stored in the Data Dictionary for the file. The following figure shows the report window for dictionary information on a single file.

File ftdb.salesrep

Field Name	Data Type	Ext	Format	Indexed
hire-date	date	0	99/99/99	No
sales-rep	character	0	!(3)	Yes
slsname	character	0	x(30)	No
slsquota	decimal	0	->,>>9.99	No
slsrgn	character	0	x(8)	No
slstitle	character	0	x(30)	No

Press space bar to continue.

Figure 7-26: Dictionary Information on One File

7.7.4 Cross Reference of Field Usage

The `Cross Reference of Field Usage` option on the Development Reports menu produces a report that lists each field in the database and the names of the forms and reports where the field appears. The following figure shows the report window for cross referencing field usage.

```
                    Cross Reference of Field Usage

Database/File/Field          Used In  Name    / Database
-------------------          -------  ----
ftdb
-------------------

agedar
     ar_inv                  Report   ar07
                             Report   ar08
     ar_invdat               Report   ar07
                             Report   ar08
     ar_lastpay              Report   ar08
     ar_pay                  Report   ar07
                             Report   aro8

customer
     Address                 Form     fm01
     Address2                Form     fm01
     City                    Form     fmo1
     Contact                 Form     fm01
     Curr-bal                Form     fm01
     Cust-num                Form     fm01
                             Report   ar04
                             Report   ar05
                             Report   ar06
                             Report   oe06
     Discount                Form     fmo1
     Max-credit              Form     fm01
     Mnth-sales[10]          Form     fmo1

Press space bar to continue.
```

Figure 7-27: Cross Reference of Field Usage

7.7.5 Field Usage in Forms

The `Field Usage in Forms` option on the Development Reports menu produces a report that lists each field in the database. In addition, it lists field names and titles as they appear in your application. The following figure shows the report window for the `Field Usage in Forms` option.

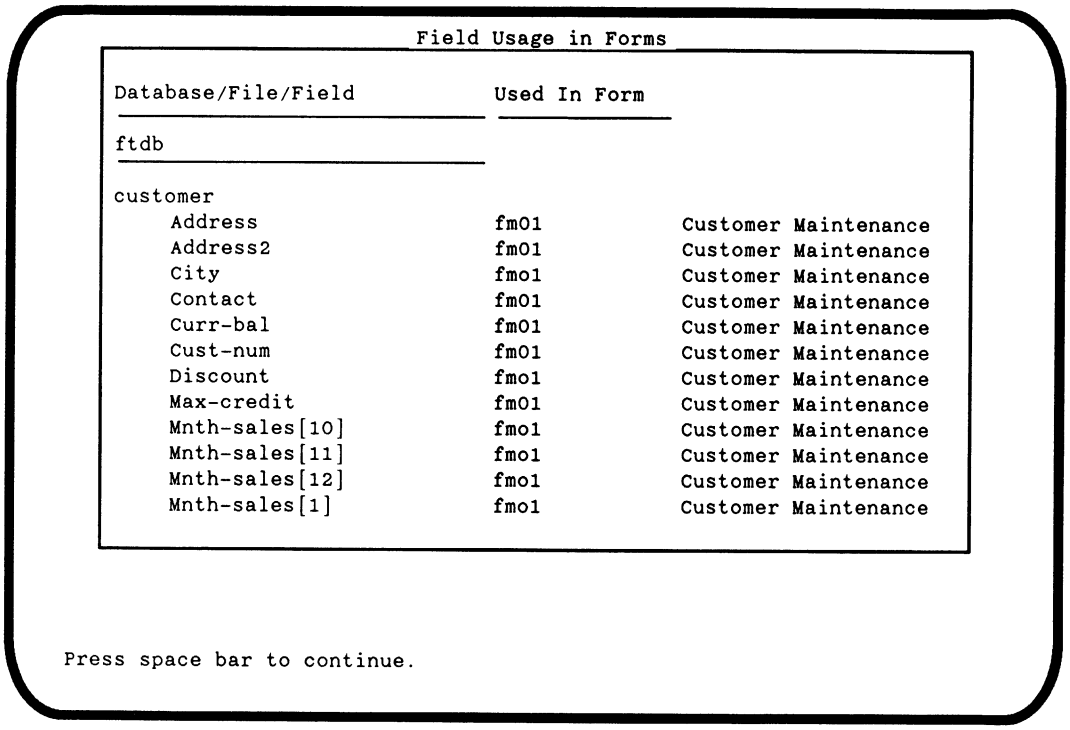
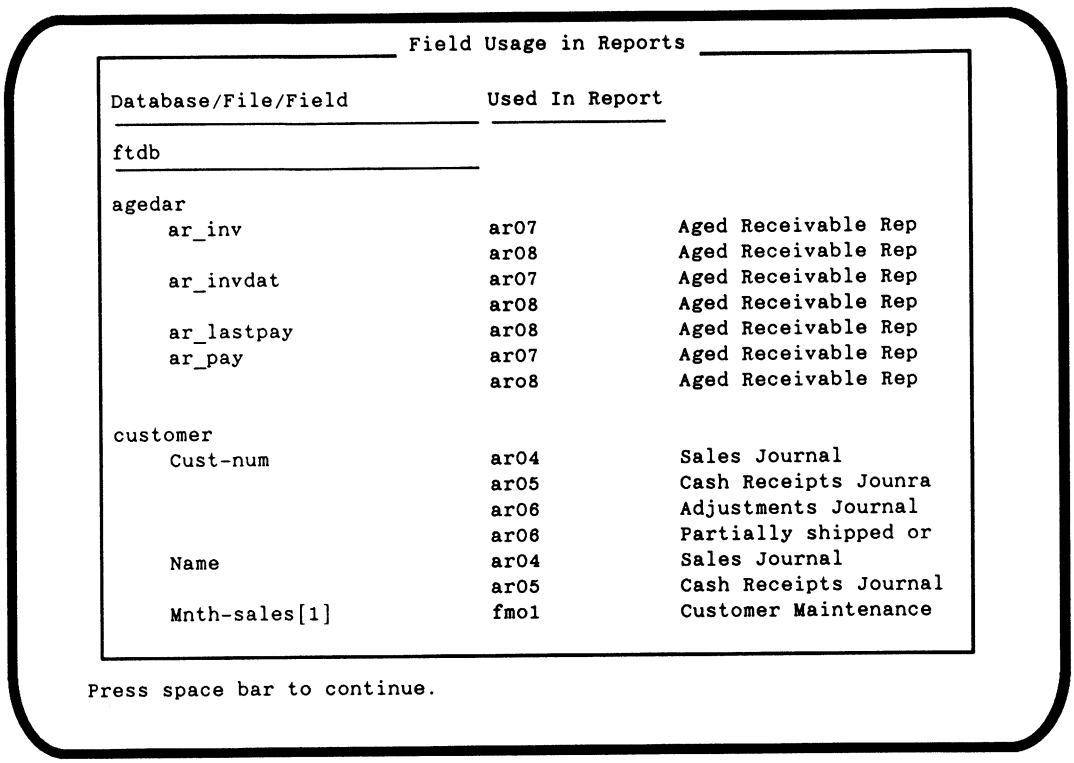


Figure 7-28: Field Usage in Forms

7.7.6 Field Usage in Reports

The `Field Usage in Reports` menu choice produces a report that lists each field in the database which has been used in a report, and lists the reports in which they are used, as well as the title of the report. Figure 7-29 shows the report window for the `Field Usage in Reports` option.



Field Usage in Reports

Database/File/Field	Used In Report	
ftdb		
agedar		
ar_inv	ar07	Aged Receivable Rep
	ar08	Aged Receivable Rep
ar_invdat	ar07	Aged Receivable Rep
	ar08	Aged Receivable Rep
ar_lastpay	ar08	Aged Receivable Rep
ar_pay	ar07	Aged Receivable Rep
	aro8	Aged Receivable Rep
customer		
Cust-num	ar04	Sales Journal
	ar05	Cash Receipts Jounra
	ar06	Adjustments Journal
	ar06	Partially shipped or
Name	ar04	Sales Journal
	ar05	Cash Receipts Journal
Mnth-sales[1]	fmo1	Customer Maintenance

Press space bar to continue.

Figure 7-29: Field Usage in Reports

7.7.7 Menu Information

When you create a menu, the Menu Editor assigns a unique identifying number to the menu. In addition, it adds a record to the database containing the Menu Settings defined for the menu and the menu's number.

The Menu Editor also adds a record to the database for each menu choice that you define. This record contains the name of the menu where this choice appears, its position within the menu (first, second, and so on), and the action type and the name assigned to the action.

You can see the database information created for your menus and menu choices with the development report produced by the `Menu Information` choice. When you select the `Menu Information` option from the Development Reports menu, FAST TRACK asks you if you want to see the menu choices. If you type **no**, a menu appears that is similar to the one in the following figure.

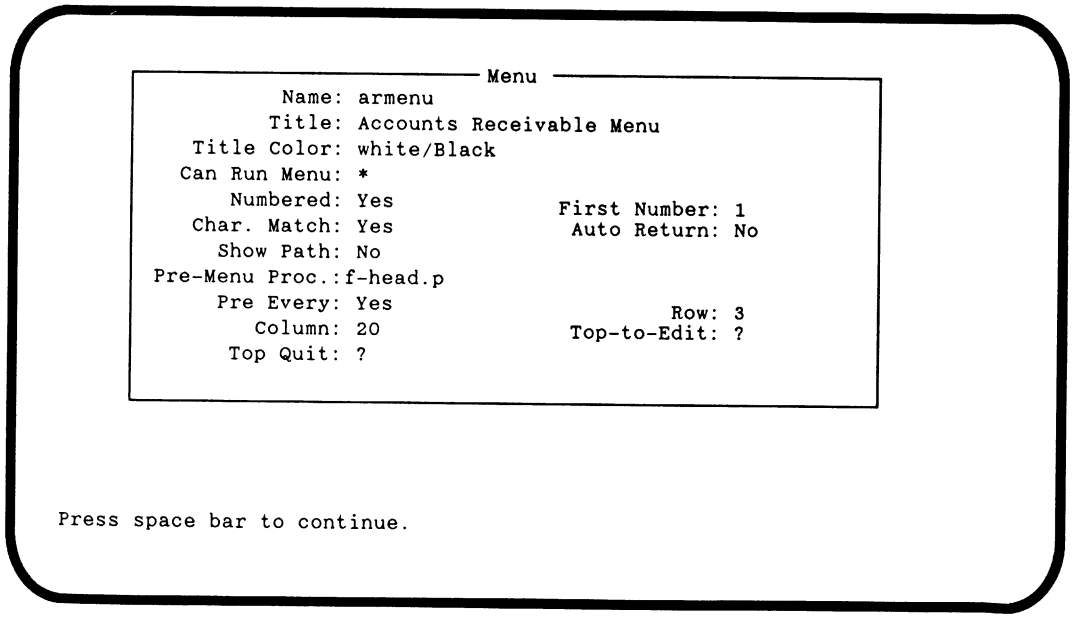


Figure 7–30: Menu Information

If you answer **yes**, FAST TRACK displays a window that contains information about each choice in the menu. Figure 7–31 shows the report window for the Menu Choices option.

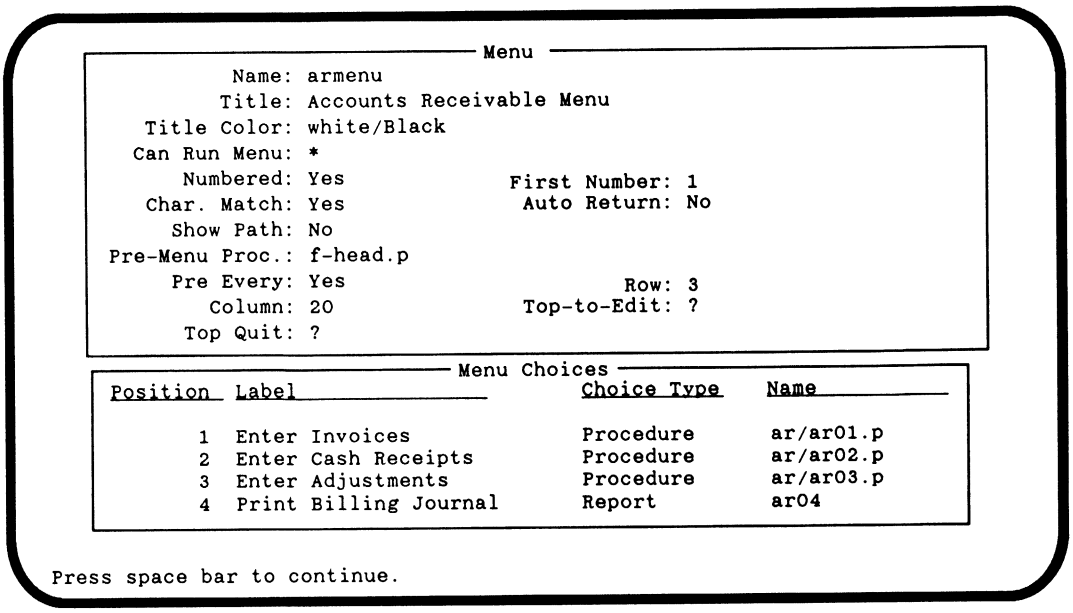


Figure 7–31: Menu Choices

Press the SPACEBAR when you have finished. FAST TRACK displays the Menu information for the next menu. To cancel the menu information display, press END (F4).

7.7.8 Report Information

Each time you create a report, the Report Writer adds a record to the database containing the report name, the name of the PROGRESS procedure you use to run your report and the report title. You can see this information in the development report generated by the `Report Information` choice. Figure 7–32 shows the report window for the `Report Information` option.

```
         Name: ar04
PROGRESS File: ar/ar04.p
        Title: Sales Journal


         Name: ar05
PROGRESS File: ar/ar05.p
        Title: Cash Receipts Journal

         Name: ar06
PROGRESS File: ar/ar06.p
        Title: Adjustments Journal




 Press space bar to continue.
```

Figure 7–32: Report Information

7.7.9 Form Information

Each time you create a form, the Screen Painter adds a record to the database containing the form name, the name of the QBF procedure file (if you generate a QBF procedure), the name of the include file (if you create an include file for the form), the position of field labels (top, left, or heading), and the unique identification number assigned to the form. You can see this information in the development report generated by the `Form Information` choice. The following figure shows the report window for the `Form Information` option.

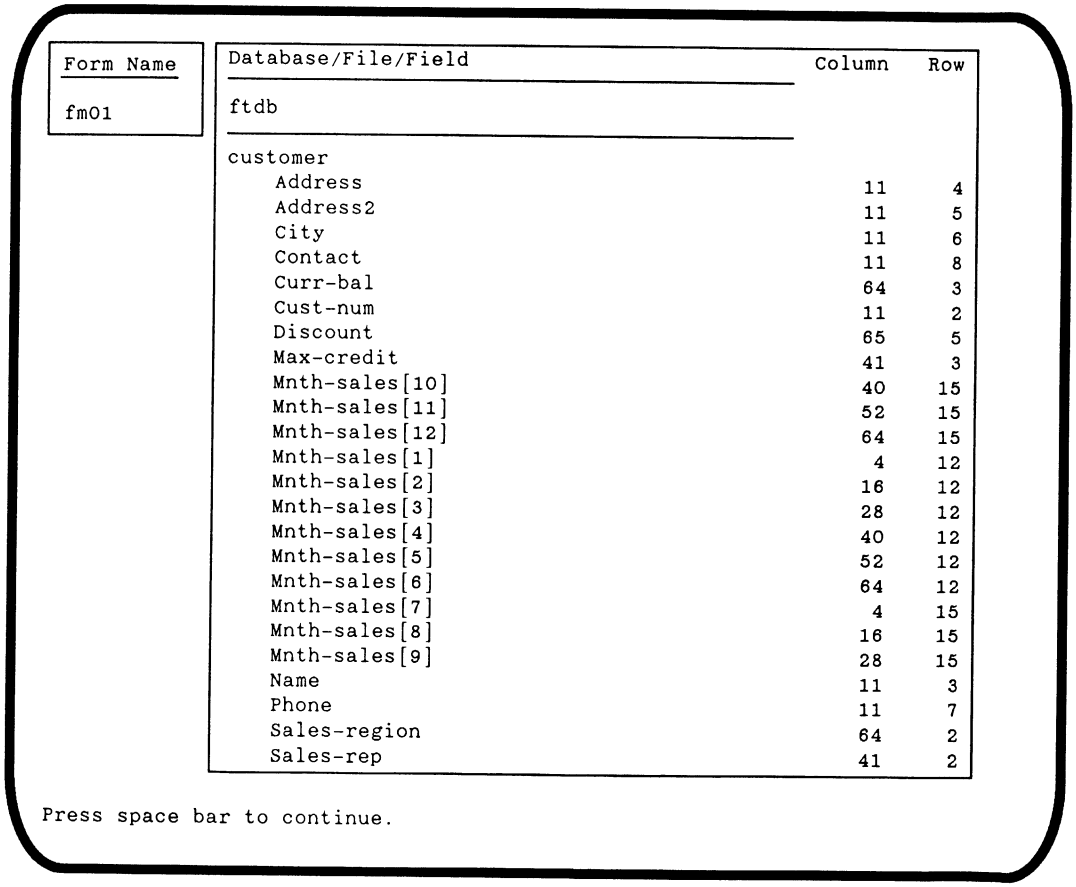
```
        Name: fmo1              Title: Customer Maintenance
        Type: left          QBF file: fm/fm01.p
Include file:

    This form is used in QBF fm01 (Customer Maintenance)


        Name: fmo2              Title: Item File Maintenance
        Type: left          QBF file: fm/fm02.p
Include file:

    This form is used in QBF fm02 (Item-maintenance)

        Name: order             Title: order
        Type: left          QBF file: fm/order.p
Include file:

Press space bar to continue.
```

Figure 7-33: Form Information

7.7.10 Field Information in Each Form

Each time you add a field to a form, the Screen Painter adds a record to the database. This record contains the name of the form in which this field appears, the name of the file that this field comes from, the name of the field, and the column and row position of the field on the form. You can see this information in the development report generated by the `Field Information in Each Form` choice.

The following figure shows the report window for the `Field Information in Each Form` option.



Form Name
fm01

Database/File/Field	Column	Row
ftdb		
customer		
Address	11	4
Address2	11	5
City	11	6
Contact	11	8
Curr-bal	64	3
Cust-num	11	2
Discount	65	5
Max-credit	41	3
Mnth-sales[10]	40	15
Mnth-sales[11]	52	15
Mnth-sales[12]	64	15
Mnth-sales[1]	4	12
Mnth-sales[2]	16	12
Mnth-sales[3]	28	12
Mnth-sales[4]	40	12
Mnth-sales[5]	52	12
Mnth-sales[6]	64	12
Mnth-sales[7]	4	15
Mnth-sales[8]	16	15
Mnth-sales[9]	28	15
Name	11	3
Phone	11	7
Sales-region	64	2
Sales-rep	41	2

Press space bar to continue.

Figure 7-34: Field Information in Each Form

7.8 UPDATE FAST TRACK DATABASE NAMES

When you create a FAST TRACK object (QBF, Report, etc.), you typically select files from different databases. These database-names (logical names) are stored in your FAST TRACK database. Every time you want to modify these objects or run a procedure, you have to have the same databases connected (same logical names and same schema). Suppose you want to change your database configuration, you want to move files between databases. Now you have to change the name of the databases in your FAST TRACK database. You can do this with the `Update FT Database Names` option. At the same time you can change the file names.

When you choose the `Update FT Database Names` option from the FAST TRACK Maintenance menu, FAST TRACK displays the following screen called "Read Me First", which explains the option.

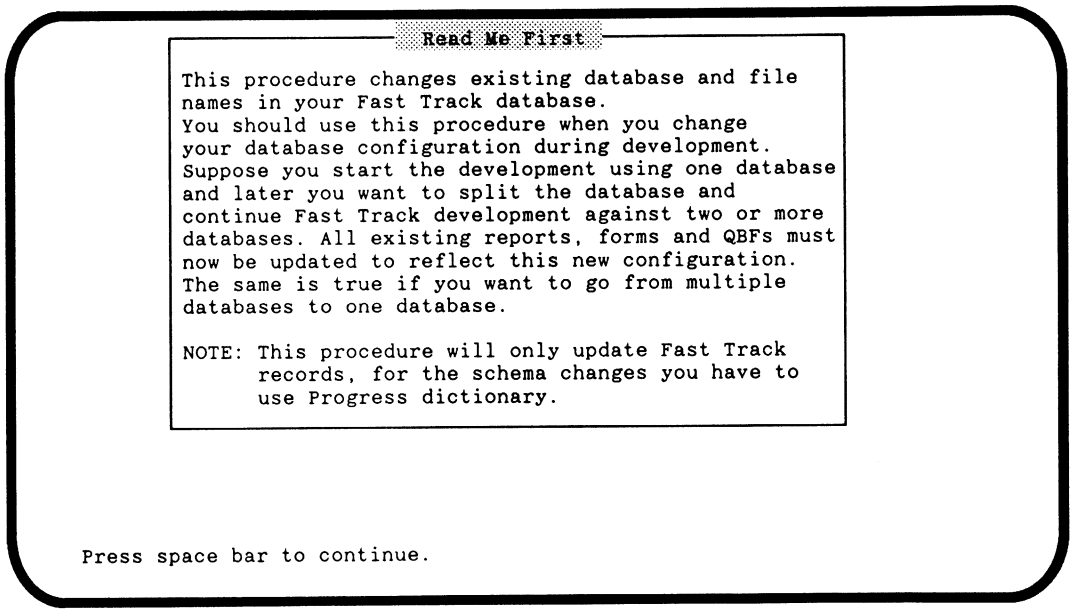


Figure 7-35: Read Me First Screen

After you read the information, press the SPACEBAR. FAST TRACK diplays the screen shown in Figure 7-36.

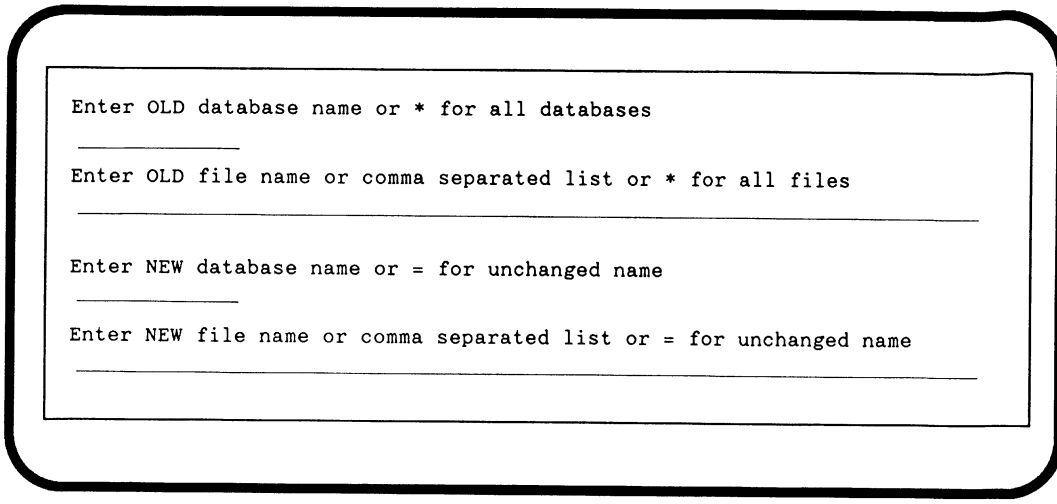


Figure 7-36: Update FT Database Names Screen

From this screen, you may do any of the following:

- Move all files in one database to another database, files kept the same names.
- Move one file in one database to another database, file kept the same name.
- Move one file in one database to another database, file was renamed.
- Move more than one file in one database to another database, files were renamed.
- Move one file in all databases to one file in one database.
- Change the name of a file in a database to another name.

Chapter 8
Multi-Database Programming With FAST TRACK

This chapter provides an introduction to "multi-database" programming with PROGRESS FAST TRACK and the PROGRESS 4th generation language (4GL). This chapter discusses:

- Database configurations required to develop and run multi-database FAST TRACK applications.
- How to start FAST TRACK with multiple databases.
- How to use FAST TRACK with multiple databases.
- Database names within FAST TRACK code and objects.

All of the concepts and information supplied in this chapter build upon those presented in Chapter 13 of the *Programming Handbook*.

8.1 INTRODUCTION TO MULTI-DATABASE PROGRAMMING WITH FAST TRACK

Multi-database programming entails the use of several databases in a single PROGRESS application. When you startup a PROGRESS session, you must provide a *connection* between the session and the database. PROGRESS provides the ability to connect to several databases from a single session. FAST TRACK is a PROGRESS application. When you startup FAST TRACK, you are starting a PROGRESS session and therefore you can also connect to multiple databases.

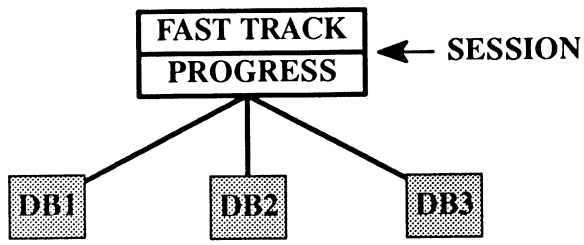


Figure 8–1: A Multi-database Application

With this capability, you can use FAST TRACK to develop reports that join data from different databases and QBF procedures that update several databases simultaneously. You can use FAST TRACK to develop multi-database applications that can simultaneously access PROGRESS, ORACLE, and RMS databases located on different operating systems using several different networking protocols.

8.2 DATABASE CONFIGURATIONS

In order to run the PROGRESS FAST TRACK product, a FAST TRACK database must be connected with "ftdb" as an alias or logical name. (A FAST TRACK database is a PROGRESS database with FAST TRACK system files defined in the schema). This database is called the *primary FAST TRACK database*. FAST TRACK uses this database to store and access FAST TRACK report, menu, form, and QBF object definitions and data files.

The primary FAST TRACK database should be distinguished from the *application databases* which contain the data used by FAST TRACK reports, forms, and QBF procedures. The primary FAST TRACK database can also be an application database. Application databases can be PROGRESS, non-PROGRESS, or FAST TRACK databases. The following diagram displays the basic database configuration for developing or running a multi-database FAST TRACK application.

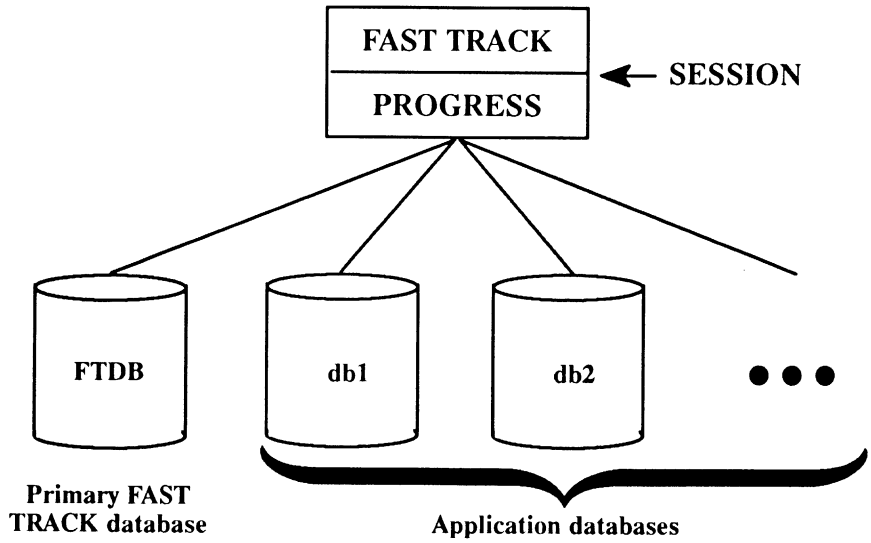


Figure 8-2: Database Configuration

The first database connected during a PROGRESS session containing an _menu file automatically receives the alias ftdb. The first database specified on the command line after a FAST TRACK startup command (proft or mproft) is connected with the logical name ftdb by default.

During the development of a multi-database FAST TRACK application, the primary FAST TRACK database should be connected with the logical name `ftdb`. When you compile a FAST TRACK multi-database application, references to data files in the primary FAST TRACK database for FAST TRACK security or output routing information receive `ftdb` as the database prefix.

To run FAST TRACK generated menus, QBF procedures, reports that use FAST TRACK output routing, and procedures that use FAST TRACK security, you must have a primary FAST TRACK database connected with the logical name or alias `ftdb`. All application databases accessed by your FAST TRACK application should be connected with the same logical database names used to generate your FAST TRACK report, form and QBF code.

For information about database connections and connection considerations, see Chapter 13 in the *Programming Handbook*.

8.3 STARTING FAST TRACK WITH MULTIPLE DATABASES

The FAST TRACK startup commands (`proft` and `mproft`) provide for the use of connect parameters on the command line. *Connect parameters* determine how you connect a session to a database. The most important connect parameter for multiple database programming is the Database Name (`-db`) parameter, which allows you to specify multiple databases on the command line at FAST TRACK startup. For example:

```
proft mydb1 -db mydb2 -db mydb3
```

In order to retain compatibility with earlier versions of PROGRESS, it is not necessary to use the Database Name (`-db`) connect parameter with the first database specified after the FAST TRACK single user and multi-user startup commands (`proft` and `mproft`). The first database specified on the command line after the FAST TRACK single user startup command (`proft`) is connected in the single user connection mode by default. All subsequent databases specified on the startup command line are connected in the default multi-user connection mode for the current machine.

The default number of databases that can be connected during a PROGRESS session is 5. The Maximum Databases (`-h`) startup parameter allows you to set the number of connected databases allowed during a PROGRESS session up to a maximum of 240. For more information about PROGRESS startup and connect parameters, see Chapter 3 in the *System Administration II: General* book.

You can group connect parameters in an ASCII file called a *parameter file*. Use the Parameter File (`-pf`) connect parameter to invoke a parameter file with a FAST TRACK startup command. For example:

```
mproft -pf parm1.pf -pf parm2.pf
```

If you enter a non-unique file name in response to a FAST TRACK prompt, FAST TRACK displays a Choices window to allow you to choose the exact file. For example, if you enter a non-unique file name into the `File Name` field in the Input Files window of the Report Writer, FAST TRACK displays a Choices window containing a list of all instances of the file name to choose from:

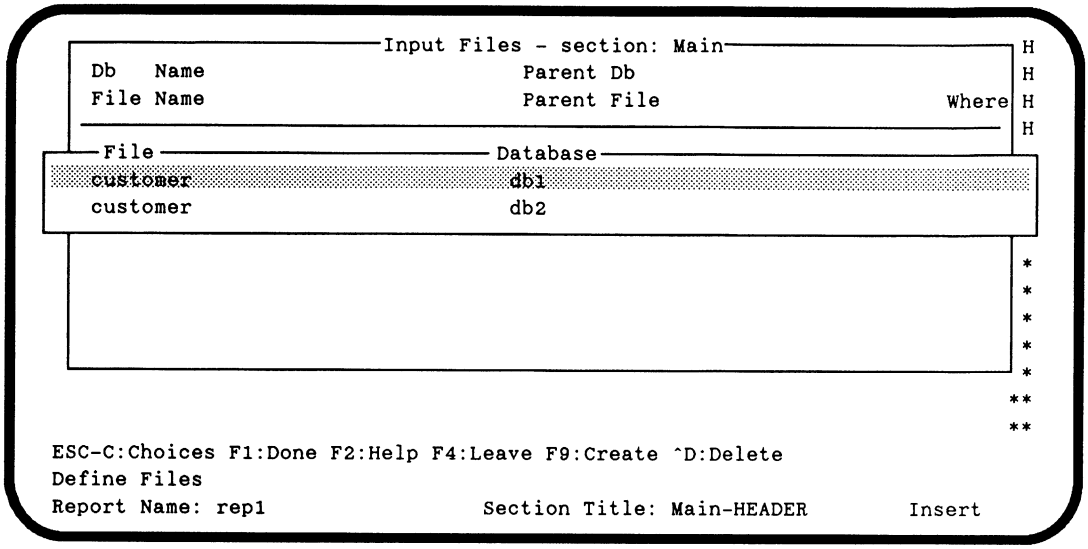


Figure 8–5: Input File Window – Report Writer

8.5 DATABASE PREFIXES

Multi-database programming with FAST TRACK is centered around database prefixes in FAST TRACK objects and generated code. *Database prefixes* are logical database names used in PROGRESS code or FAST TRACK object definitions to qualify file and field references. The following sections discuss database prefixes in relation to code and objects generated by FAST TRACK.

8.5.1 Database Prefixes and FAST TRACK Generated Code

When you generate code for forms, reports, and QBF procedures with FAST TRACK, you have the option of including database prefixes for file references in the generated code. For example, if your object contains a reference to a database file called `customer`, which is located in a database connected with the logical database name `demo1` and you choose to generate code with database prefixes, all references to the `customer` file appear in the code as follows:

```
demo1.customer
```

Database prefixes are necessary to resolve ambiguous file references in instances where your object contains references to non-unique file names.

If all of the file references in your object are unique, you may not want to generate code with database prefixes. Without database prefixes in your generated code, you have the freedom to change the database configuration and the location of files referenced in the code without having to recode your file references. All you have to do is recompile your code.

When you select the COMMAND→GENERATE command to generate form code in the Screen Painter or report code in the Report Writer, FAST TRACK displays a prompt asking if you want database prefixes in the generated code. For example, the following prompt appears when you select the COMMAND→GENERATE command in the Report Writer:

```
Command Generate
Report Name: rep1                Section Title: Main-HEADER          Insert
Do you want database prefix on file names ? yes
```

Figure 8–6: Database Prefix Prompt – Report Writer

This prompt is similar to the prompt that occurs in the Screen Painter when you generate form code. If you answer **yes** to this prompt, the appropriate database prefixes are added to the file references in your generated code. If you answer **no** to the prompt and there are non-unique file names in your form or report object, a compilation error occurs and FAST TRACK displays a messages telling you that there are ambiguous file references in your generated code. For example:

```
Your report program is written, now compiling ...
Command Generate
Report Name: rep1                Section Title: Main-HEADER          Insert
file name customer is in database db1 and db2
Unknown or ambiguous file customer
Press space bar to continue.
```

Figure 8–7: Ambiguous File Message – Report Writer

When you generate QBF procedures with the QBF Generator, the `Add database prefix in the gen. code` field allows you to specify whether or not you want database prefixes in the generated QBF code.

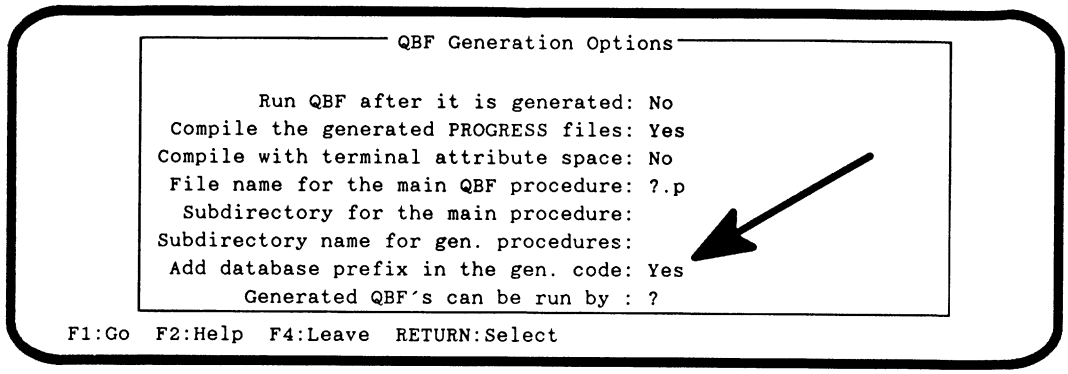


Figure 8–8: QBF Generation Options Window

If you answer **yes** to this field, the appropriate database prefixes are added to the file references in your generated code. If you answer **no** to this field and you are generated QBF procedures for files with non-unique file names, a compilation error occurs and message appears notifying you of an ambiguous file reference.

8.5.2 Database Prefixes and FAST TRACK Generated Objects

All report, form, and QBF objects in the primary FAST TRACK database contain database prefixes for file references, even if you generated corresponding code without database prefixes. If you want to edit an existing object with one of the FAST TRACK editors, you must have databases connected with the same logical database names that are referenced in the object that you want to edit. If you do not have the proper logical names established prior to entering an editor to edit an object, FAST TRACK displays a message informing you that a particular logical database name is not connected and denies editor access to the object.

The `Update FT Database Names` option on the FAST TRACK Maintenance Menu allows you to change all instances of a database prefix and/or file name in objects in your primary FAST TRACK database to a new database prefix and/or file name.

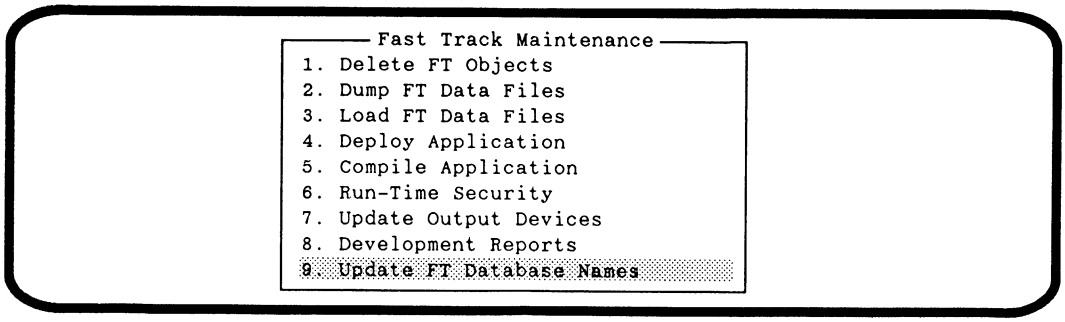


Figure 8–9: FAST TRACK Maintenance Menu

Use this command during the development process, if you change your database configuration, a logical database name, or the location of a file. You must regenerate any procedures affected by the database prefix or file name change.

When you select the Update FT Database Names option from the FAST TRACK Maintenance Menu, FAST TRACK displays a message explaining the menu option and then displays the Change Database Prefix window:

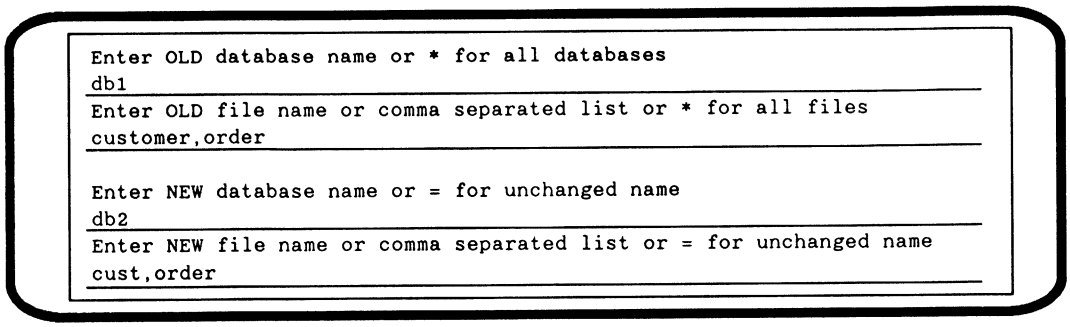


Figure 8–10: Change Database Prefix Window

The example in the figure above shows a database prefix change (db1 to db2) for two files (customer and order). The customer file has a different name (cust) in the db2 database. The name of the order file is the same in both databases. When you execute the database prefix modification based upon the entries in the window above, all db1.order file references in the FAST TRACK objects of the primary FAST TRACK database change to db2.order. All db1.customer file references in the FAST TRACK objects of the primary FAST TRACK database change to db2.cust.

Before executing the changes specified on the Change Database Prefix window, FAST TRACK displays a summary report of the intended database prefix and file name changes and prompts you to confirm the modification:

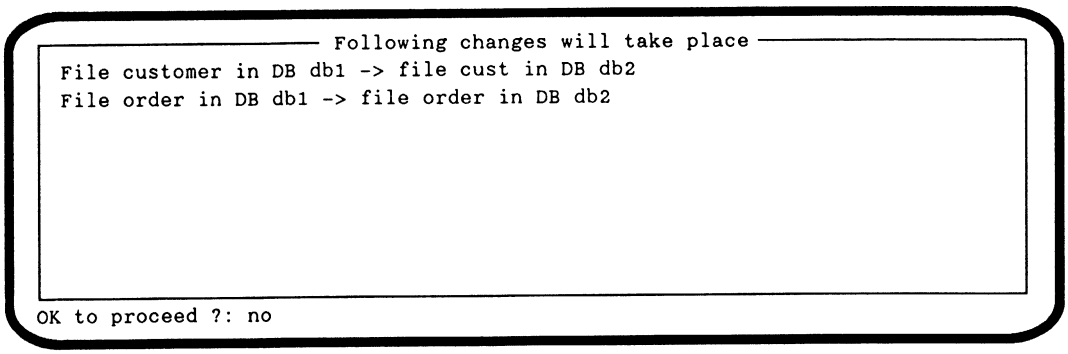


Figure 8–11: Database Prefix Modification Report And Prompt

If you answer **yes** to the prompt, FAST TRACK replaces all instances of the old database prefix in your FAST TRACK objects with the new database prefix. FAST TRACK displays messages informing you of the files affected by the database prefix modification.

8.5.3 Database Prefixes And conv56

To convert a Version 5 FAST TRACK database to a Version 6 FAST TRACK database, use the **conv56** option of the `proutil` command. Chapter 4 of *System Administration II: General* provides a detailed explanation of the `proutil` command.

Version 5 FAST TRACK does not support multi-database programming, therefore there are no database prefixes in the objects and code generated with the Version 5 FAST TRACK product. When you run the Version 6 FAST TRACK product for the first time on newly converted database, FAST TRACK automatically inserts the database prefix `ftdb` for all file references in the existing FAST TRACK objects. Prior to displaying the FAST TRACK Main Menu, the following message appears:

```
All existing forms, QBF's and reports without db-name will get db-name FTDB.

Press space bar to continue.
```

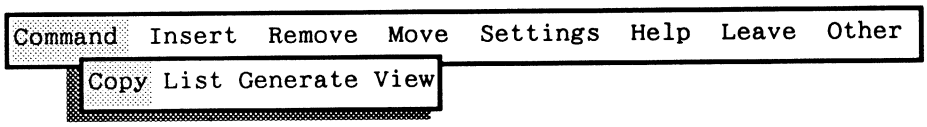
Figure 8–12: Default Database Prefix Message

Any procedures associated with FAST TRACK objects in the newly converted database must be recompiled. If you want the database prefix in your code, you must regenerate your procedures with FAST TRACK. If you convert a Version 5.2D or earlier FAST TRACK database to a Version 6 FAST TRACK database, all QBF procedures associated with the QBF objects in the newly converted database must be regenerated.

Appendix A
The Menu Editor Command Menu

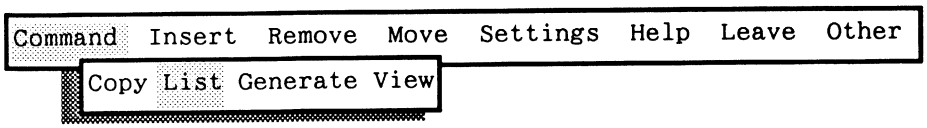
You access the Menu Editor command menu by pressing either CTRL-O, or the key sequence you have defined as the OPTIONS. The Menu Editor commands give you enhanced editing options and perform system functions for the Menu Editor such as saving and viewing files. This appendix summarizes these commands in their order of presentation on the command menu.

COMMAND→COPY



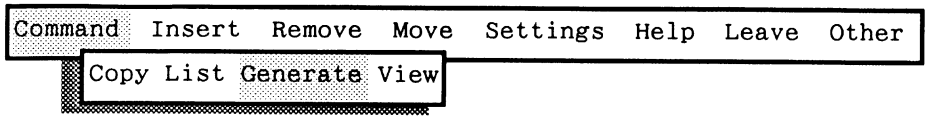
You use the COMMAND→COPY command to copy a menu. When you make this selection, FAST TRACK prompts you for the name of the menu that you wish to copy. After you enter the menu name and press GO or RETURN, FAST TRACK *replaces* everything in your current menu with the menu that you want to copy. The title of the copied menu now appears in the top border of the frame. Note, however, that the name of your menu has not changed.

COMMAND→LIST



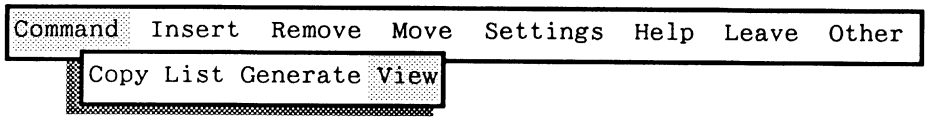
When you select this command, FAST TRACK displays an indented listing that outlines the current state of your menu system. This listing shows all menu choices for each menu in your system and the type of action associated with that choice. This listing also indicates whether a choice type has been defined for each menu choice.

COMMAND→GENERATE



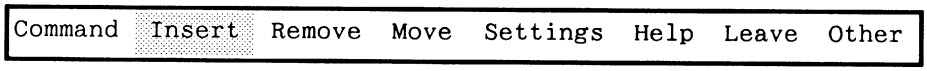
You use the COMMAND→GENERATE command to generate a PROGRESS procedure file for your menu. When you choose this command, FAST TRACK prompts you for the name you want to give to the file.

COMMAND→VIEW



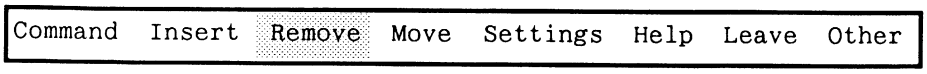
You use the COMMAND→VIEW to test a menu. When you use this command, FAST TRACK displays the menu as it appears in your application.

INSERT



You use the INSERT command to insert a choice line into your menu. To use the INSERT command, highlight the menu choice line at the position where you want to insert the new line. When you press [OPTIONS] ([CTRL]-[O]), FAST TRACK moves the currently highlighted choice down and renumbers it. If you want to insert a choice after the current line, press [MODE] (F3) to change to insert mode. Then press [RETURN].

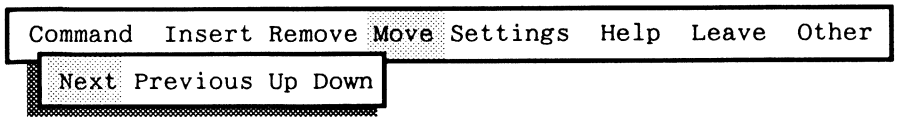
REMOVE



You use the REMOVE command to remove a choice from a menu. To use this command, highlight the choice that you want to remove and press [OPTIONS] ([CTRL]-[O]). FAST TRACK removes the choice, renumbers the following choices (if they were numbered) and moves them up.

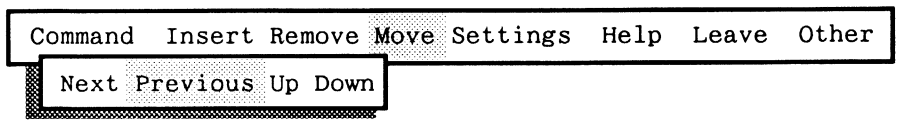
The MOVE commands are used to highlight a menu choice for editing.

MOVE→NEXT



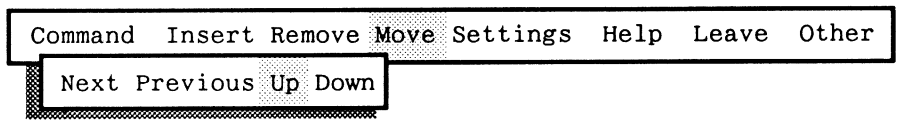
You use the MOVE→NEXT command to highlight the next choice in your menu. If the last choice in your menu is highlighted, the MOVE→NEXT command selects the first choice in the menu. Note that the ↓ key also highlights the next choice in a menu.

MOVE→PREVIOUS



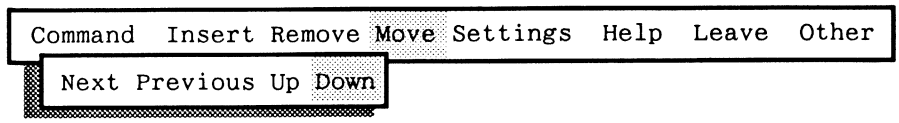
You use the MOVE→PREVIOUS command to highlight the previous choice in a menu. If the first choice in a menu is highlighted, then this command selects the last choice in the menu. Note that the ↑ key also highlights the previous menu choice.

MOVE→UP



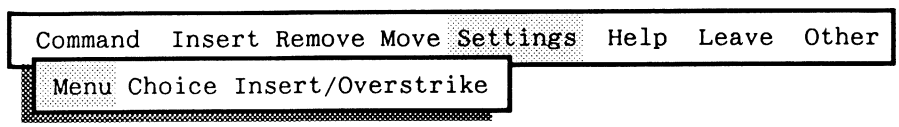
The MOVE→UP command selects the first choice in a menu.

MOVE→DOWN



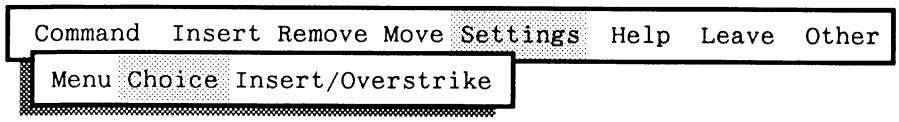
The MOVE→DOWN command selects the last choice in a menu.

SETTINGS→MENU



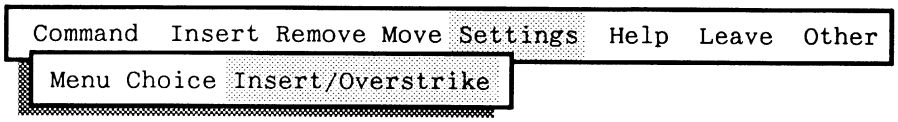
The SETTINGS→MENU command display the Menu Settings window. Invoke this option when you want to change the default settings of the FAST TRACK interface.

SETTINGS→CHOICE



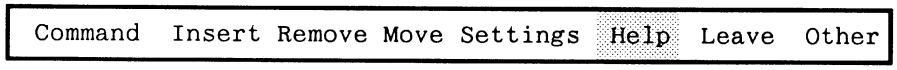
When you select SETTINGS→CHOICE, the Settings Choice menu appears. This menu allows you to define or modify an action type for the particular menu choice.

SETTINGS→INSERT/OVERSTRIKE



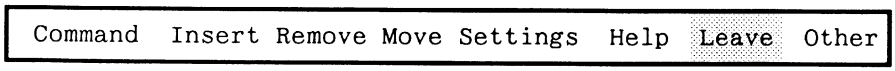
The SETTINGS→INSERT/OVERSTRIKE command allows you to toggle between insert and overstrike modes. On most systems this is equivalent to pressing [MODE] (F3).

HELP



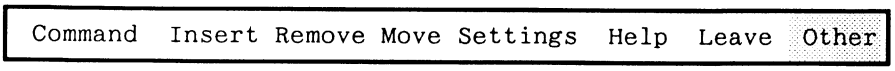
The HELP command provides help on the Menu Editor.

LEAVE



You choose the LEAVE command to leave the Menu Editor and return to the FAST TRACK Main Menu.

OTHER



The OTHER commands are covered in detail in Chapters 1 and 7 of this manual. They include OTHER→OPSYS, OTHER→DICTIONARY, OTHER→MAIN-MENU, OTHER→REPORTS, and OTHER→GOTO. These commands are common to all three FAST TRACK editors.

Appendix B
The Screen Painter Command Menu

You access the Screen Painter's horizontal command menu by pressing either CTRL-O or the key sequence you have defined as OPTIONS. The Screen Painter commands give you enhanced editing options and perform system functions for the Screen Painter such as saving and viewing files. This appendix summarizes these commands in their order of presentation on the command menu.

COMMAND→DEFAULT

```
Command Define Insert Remove Move Pick Settings  Help  Leave  Other
        Default Copy Generate View Save
```

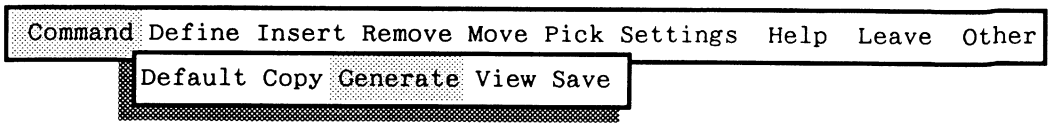
When you choose this command, FAST TRACK automatically creates a form with all of the fields in a single file. When you enter this command, FAST TRACK prompts you for the filename. Note that a default form can be created for only one file, even if you have defined multiple files for it.

COMMAND→COPY

```
Command Define Insert Remove Move Pick Settings  Help  Leave  Other
        Default Copy Generate View Save
```

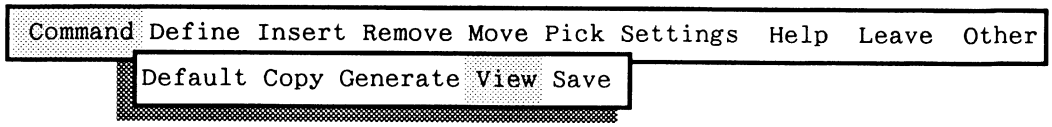
Copies a form into the current form. When you choose this option, FAST TRACK prompts you to enter the name of the form you want to copy.

COMMAND→GENERATE



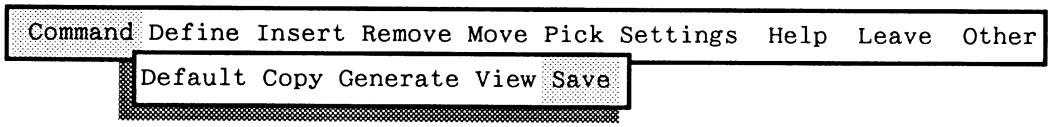
The COMMAND→GENERATE command generates an include file containing the form definition. You can use a form in conjunction with a PROGRESS procedure to manipulate data from either a FAST TRACK or a PROGRESS database.

COMMAND→VIEW



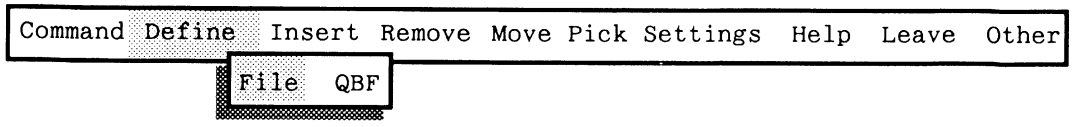
This command allows you to view your form as it will appear on the screen in your final application. If you have specified a title for your form, it appears centered in the top border of the frame.

COMMAND→SAVE



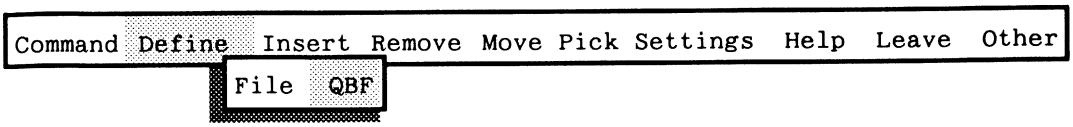
Use this command to save your form in the database. No PROGRESS procedure files are created. This command also gives you a way to make a checkpoint save.

DEFINE→FILE



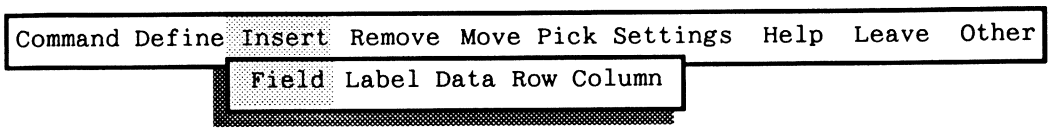
The DEFINE→FILE command makes files from your database available to the current form. You must define files, before you can insert fields from the files. After you select the DEFINE→FILE command, an Input Files window appears. You can enter up to 12 files in this window.

DEFINE→QBF



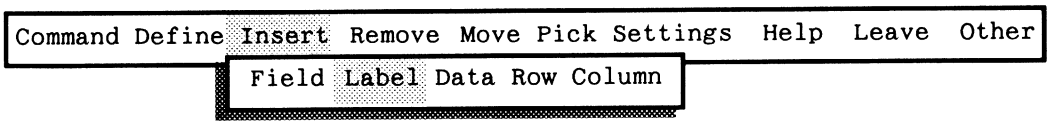
The DEFINE→FILE command creates a QBF procedure for a form, or updates the QBF for a form after you have made changes to the form. Before you can do this, however, it is necessary to insert the fields for the file into the form. When you select this command the QBF Settings window appears.

INSERT→FIELD



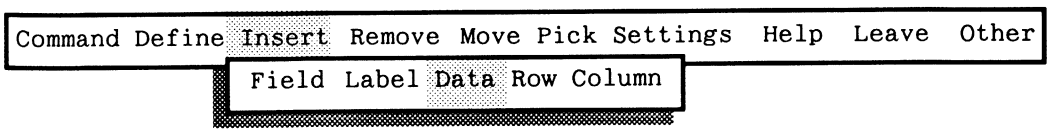
You use the INSERT→FIELD command to insert fields into your form. The number of fields in your form is limited only by how many will fit on a screen. You must define the files from which the fields are taken prior to using this command. When you choose this command, a menu of files and fields appears

INSERT→LABEL



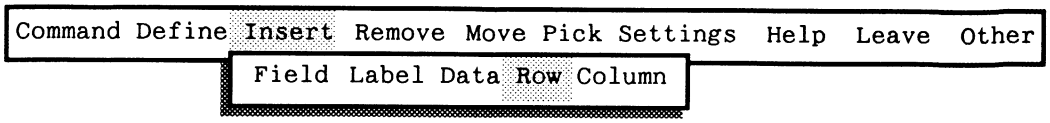
You use the INSERT→LABEL command to insert only a field's label into your form at the position in which the cursor is located. When you select this command, the files and fields choices window appears.

INSERT→DATA



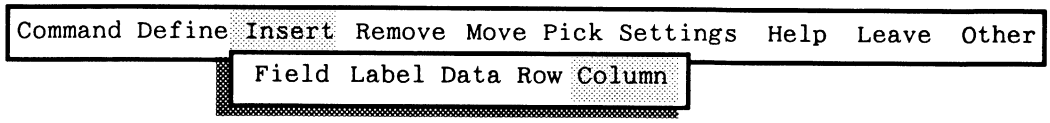
You use the INSERT→DATA command to insert only a field's data area into your form at the position at which the cursor is located. When you select this command, the files and fields choices window appears.

INSERT→ROW



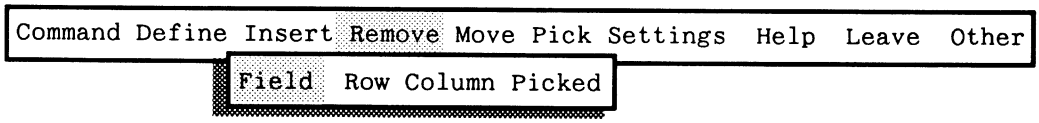
You use the INSERT→ROW command to Insert a new empty row at the position of the cursor. All rows beginning with the row on which the cursor was located are pushed one row down.

INSERT→COLUMN



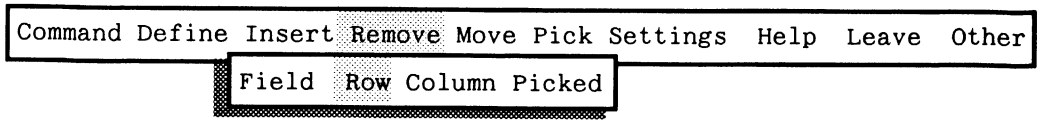
You use the INSERT→COLUMN command to insert a (vertical) column of blank spaces into your form in the column in which the cursor is located. Your cursor must be positioned in a blank column prior to selecting this command. Everything to the right of the new column is pushed one character to the right.

REMOVE→FIELD



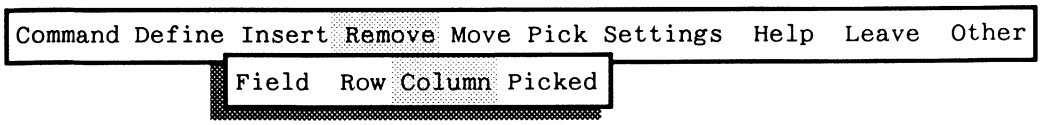
Position the cursor anywhere in a field (either label or data area). Use REMOVE→FIELD to remove the entire field from your form. If you want to put the field back in your form, you must use the INSERT→FIELD command.

REMOVE→ROW



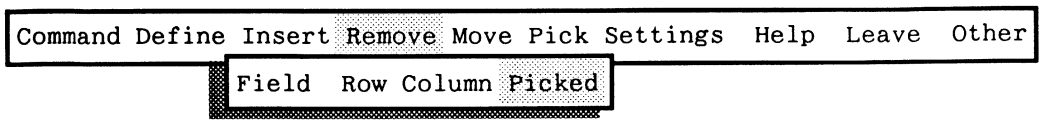
REMOVE→ROW removes a row and anything that happens to lie on it from your form. Rows below the selected row move up to fill up the space.

REMOVE→COLUMN



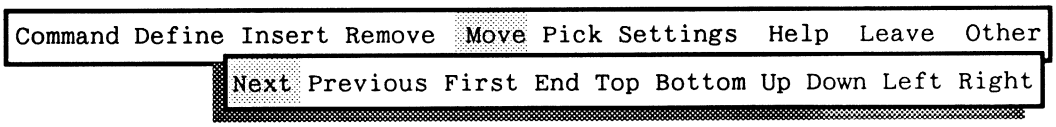
REMOVE→COLUMN removes an *empty* column from your form. If any text or data crosses the column at any point on the screen, then you cannot remove the column.

REMOVE→PICKED



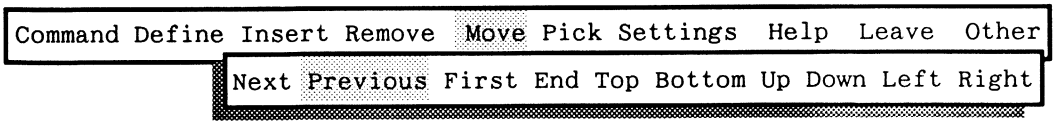
REMOVE→PICKED removes anything that has been marked by a PICK command. If you want to reinsert the removed area, use the INSERT commands.

MOVE→NEXT



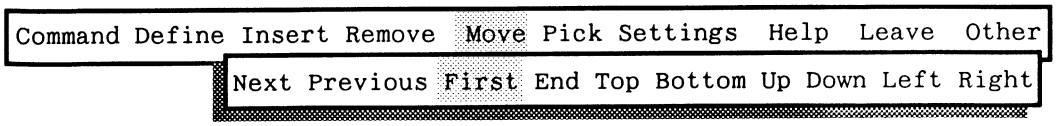
The MOVE→NEXT command moves or tabs the cursor from a field label to the data area, and from data area to the next field. If the cursor is located on the last area (either label or data) in a form, the cursor moves to the *first* field area in the form.

MOVE→PREVIOUS



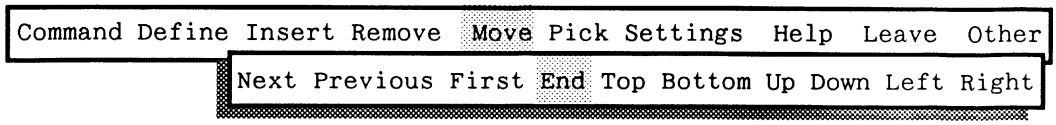
The MOVE→PREVIOUS command moves the cursor from a field data area back to the label, and from a label to the data area of the previous field in the form. If the cursor is located on the first field area in the form, then this command moves it to the last field area in the form.

MOVE→FIRST



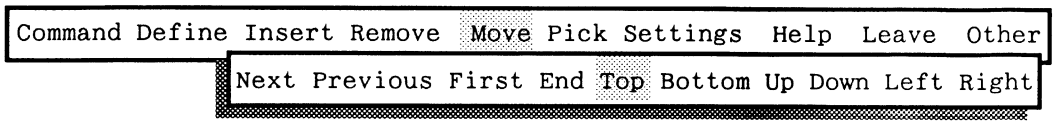
The MOVE→FIRST command moves the cursor to the first character position of the left-most field on the row on which the cursor is located.

MOVE→END



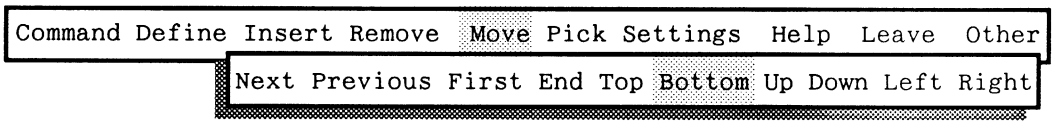
The MOVE→END moves the cursor to the last character position in the last field on the current row.

MOVE→TOP



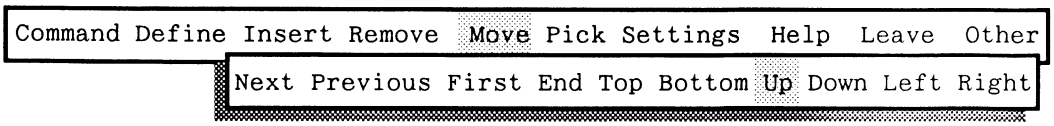
The MOVE→TOP command moves the cursor to the top-most row that has a label, data, or text in the current column.

MOVE→BOTTOM



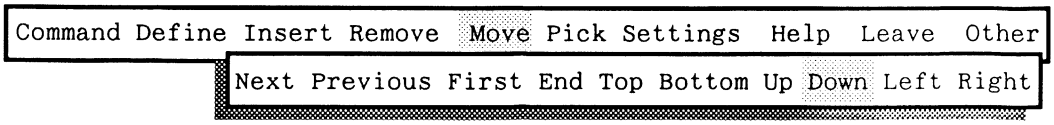
MOVE→BOTTOM moves the cursor to the bottom-most field that has a label, data, or text in the current column.

MOVE→UP



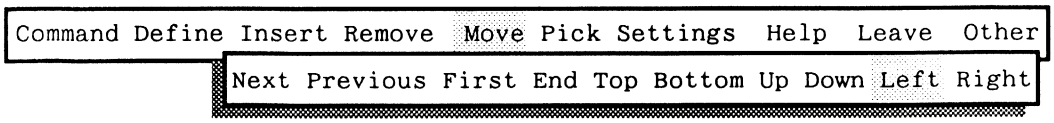
The MOVE→UP command scrolls the form up by the number of lines on a screen, for forms that are longer than the screen.

MOVE→DOWN



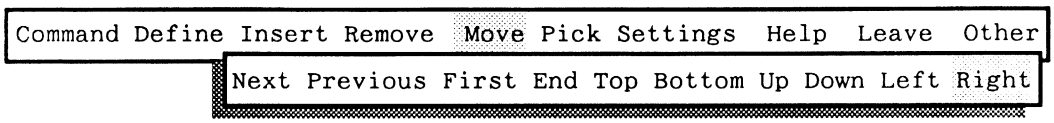
The MOVE→DOWN command scrolls the form down by the number of lines on the screen, for forms that are longer than the screen.

MOVE→LEFT



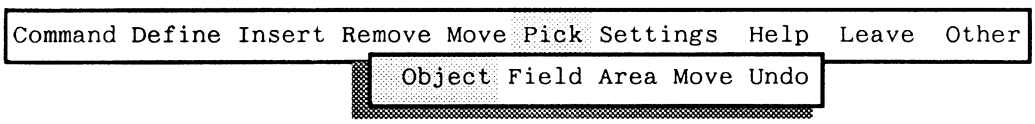
The MOVE→LEFT command scrolls 20 places to the left when the form is wider than the screen.

MOVE→RIGHT



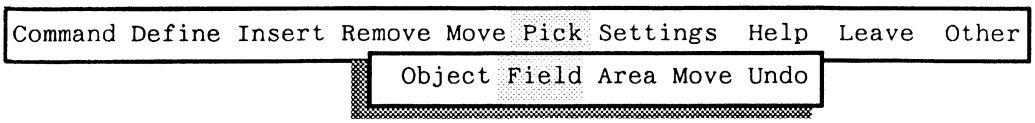
The MOVE→RIGHT command scrolls 20 places to the right when the form is wider than the screen.

PICK→OBJECT



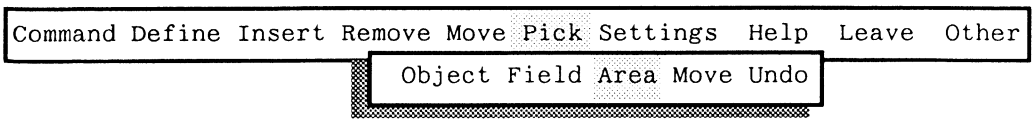
The PICK→OBJECT command lets you select and mark a field's label or data area. To use this command, first move the cursor to the field label or data area.

PICK→FIELD



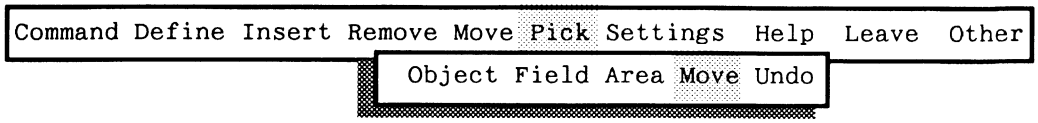
The PICK→FIELD command is used to mark an entire field (label and data area). Your cursor should be located in the field you wish to mark prior to using this command.

PICK→AREA



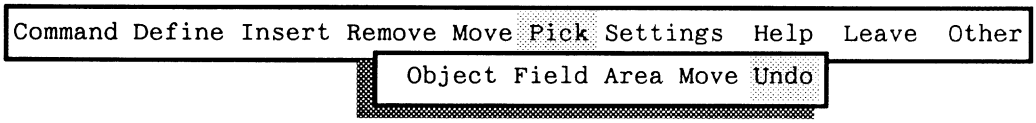
The PICK→AREA command selects a group of fields to be moved. To use this command, you must first position the cursor at any corner of the area you want to select. After you press PICK→AREA, FAST TRACK prompts you to move the cursor to the opposite corner of the area you wish to mark and press [ESC]-[W] to mark the area. (You can also choose PICK→AREA again.) FAST TRACK highlights the area that you have marked. Note that you cannot mark an area that cuts through a field label or data area: the entire label or field must fall within the area to be marked. If you attempt to mark an area that cuts through a field, FAST TRACK displays an error message and prompts you to try again.

PICK→MOVE



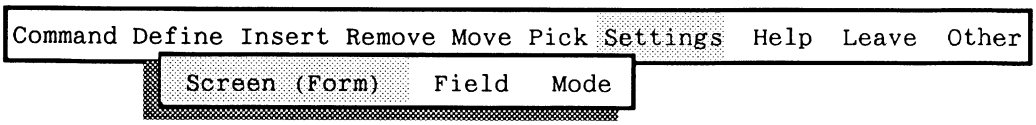
The PICK→MOVE command moves the selected field, label, data area, or group of fields. The upper left corner of the area will be moved to the position of your cursor prior to invoking this command.

PICK→UNDO



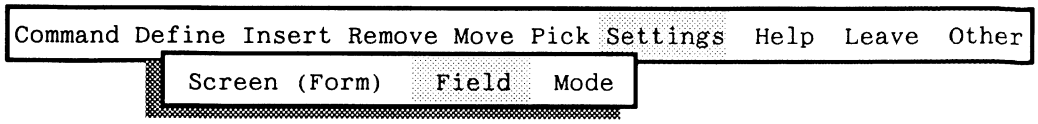
PICK→UNDO unmarks anything that has been marked.

SETTINGS→SCREEN



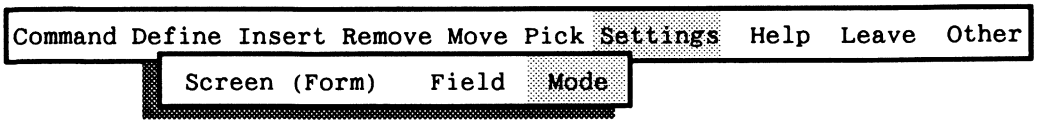
The SETTINGS→SCREEN command allows you to define characteristics of your form such as its title and size, and whether it is displayed with a box. When you choose the SETTINGS→SCREEN command, the Screen Painter displays the Screen Settings window.

SETTINGS→FIELD



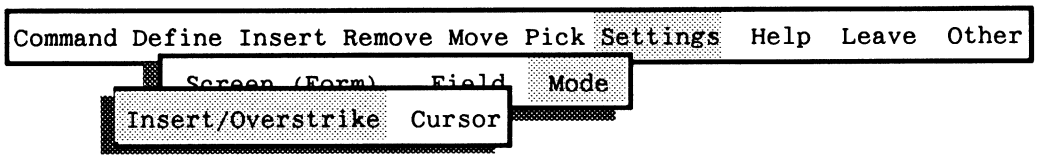
The SETTINGS→FIELD command is used to change field attributes. When you position your cursor anywhere in a field and choose the SETTINGS→FIELD command, the Field Attribute Setting window appears as in the next figure. Only the fields marked for input can be changed. For example, some systems may mark fields for input by underlining them.

SETTINGS→MODE



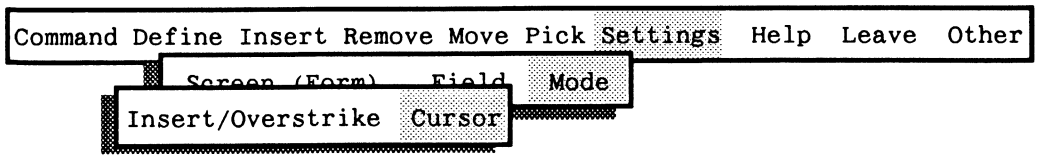
The SETTINGS→MODE command allows you to toggle between insert and overstrike modes or between cursor position display and non-display modes. modes. Refer to the next two SETTINGS commands for additional information.

SETTINGS→MODE→INSERT/OVERSTRIKE



You select SETTINGS→MODE→INSERT/OVERSTRIKE to toggle between Insert and Overstrike mode. On most systems this is synonymous with [MODE] (F3).

SETTINGS→MODE→CURSOR

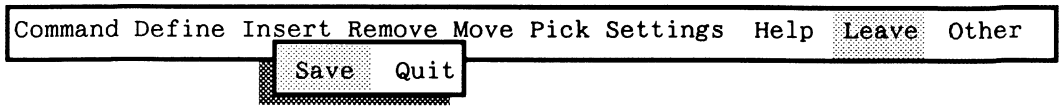


The SETTINGS→MODE→CURSOR command tells FAST TRACK to display the cursor row and column position in the lower right corner of the screen while you are creating a form. To turn off the row/column display, simply select this command again.

HELP

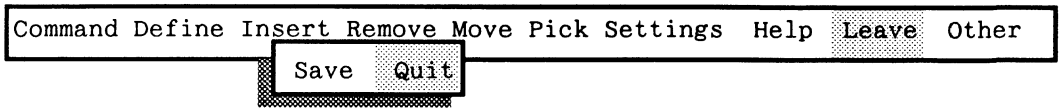
The HELP command gives you access to help information on the Screen Painter. You can also access the help facility by pressing the F2 key.

LEAVE→SAVE



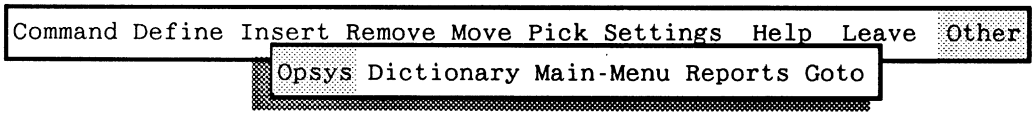
The LEAVE→SAVE command saves the formatting and data information of your current form in the current database. FAST TRACK then returns you to the Main Menu.

LEAVE→QUIT



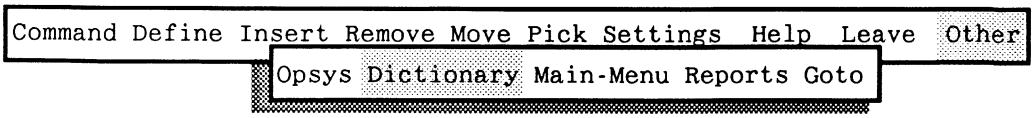
The LEAVE→QUIT command returns you to the Main Menu without saving your form. Use this command when you want to abandon your most recent changes to a form.

OTHER→OPSYS



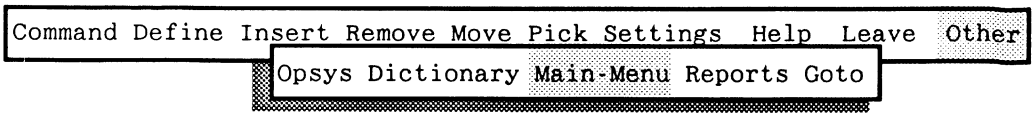
The OTHER→OPSYS command allows you to temporarily leave FAST TRACK to use the operating system. You return to the application by typing **exit** from DOS, by pressing CTRL-D from UNIX, by typing **PROGRESS Exit** from BTOS/CTOS, and by typing **logout** from VMS.

OTHER→DICTIONARY



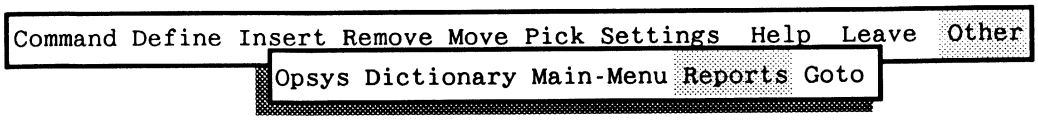
The OTHER→DICTIONARY command allows you to use the PROGRESS Data Dictionary. For information on using the Data Dictionary, refer to the *PROGRESS Language Tutorial*.

OTHER→MAIN-MENU



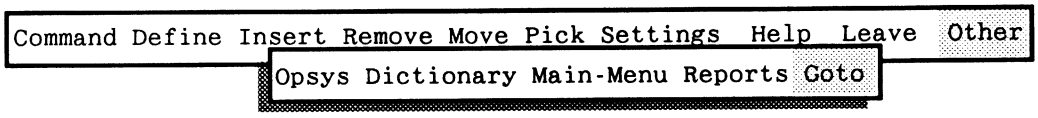
When you select the OTHER→MAIN-MENU command FAST TRACK asks you whether you want to save your changes. After you respond (either **yes** or **no**), you return to the FAST TRACK Main Menu.

OTHER→REPORTS



The OTHER→REPORTS command allows you access to information about your menus, forms, reports and QBFs. For more information on these reports, see the chapter on Development Reports in this manual.

OTHER→GOTO

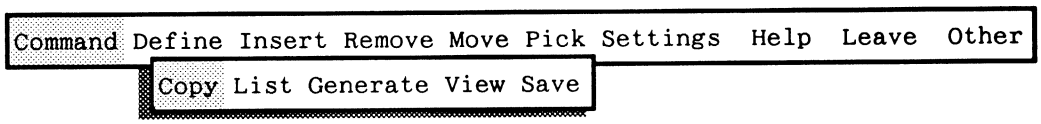


You select the OTHER→GOTO command to execute a FAST TRACK object (either a menu, QBF,report, or a PROGRESS procedure tied to a menu). When you select this command, FAST TRACK displays the Goto window.

Appendix C The Report Writer Command Menu

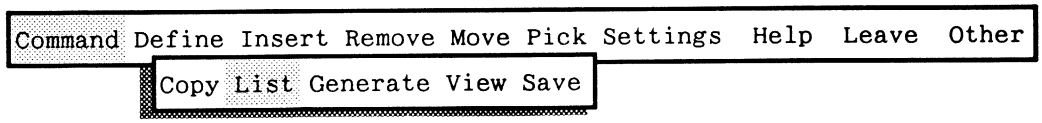
You access the Report Writer's horizontal command menu by pressing either CTRL-O or the key sequence you have mapped as OPTIONS . The Report Writer commands give you enhanced editing options and perform system functions for the Report Writer such as running reports and saving files. This appendix summarizes these commands in their order of presentation on the command menu.

COMMAND→COPY



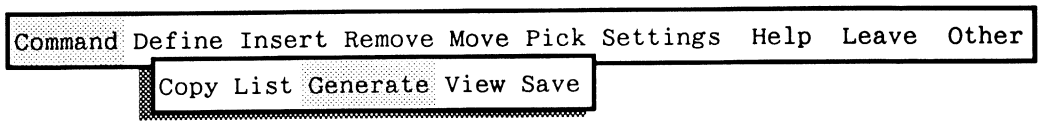
The COMMAND→COPY command allows you to use other reports as templates to create new reports. This command copies an existing report object from the current database into the current report.

COMMAND→LIST



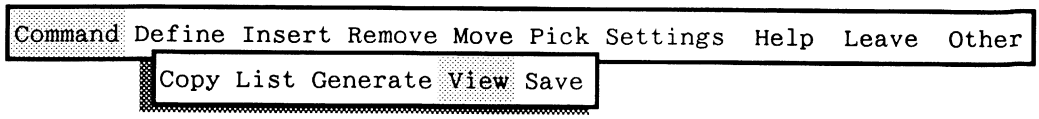
The COMMAND→LIST command displays the logical structure of the current report. The structure of a report shows report sections and how they are nested.

COMMAND→GENERATE



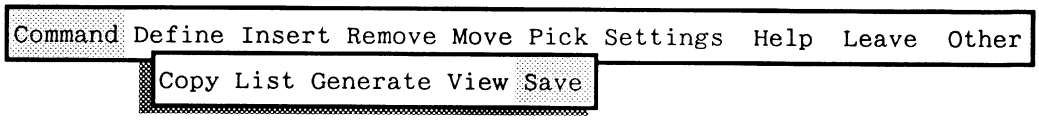
Select the COMMAND→GENERATE command to create a PROGRESS procedure for your current report. Every report for which you generate a procedure has at least one include (.i) file. The include file contains the formatting information for a particular section within your report.

COMMAND→VIEW



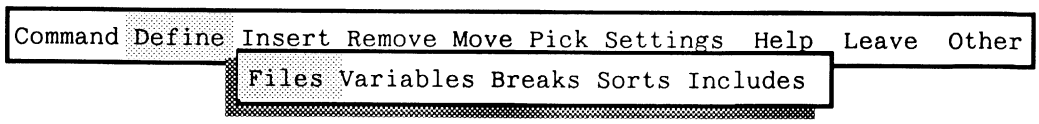
Use the COMMAND→VIEW command to generate, compile, and run a temporary report procedure and display the report output on your terminal.

COMMAND→SAVE



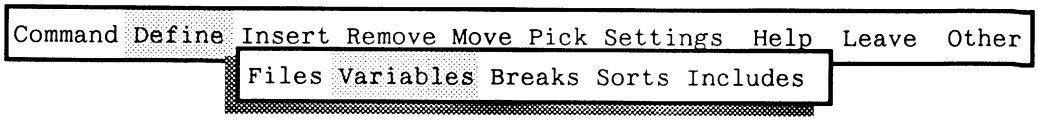
When you save your report using COMMAND→SAVE, the Report Writer saves it in the database as an object. In other words, when you save your report to the database, the report's section, file, field, and formatting information is saved in various database files. After you save a report as an object, you can retrieve the object to recreate your report or use it as a template for other reports. You can also delete objects from your current database using the Maintenance option on the FAST TRACK Main Menu.

DEFINE→FILES



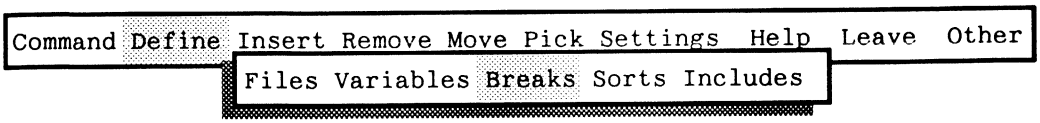
The DEFINE→FILES command makes database files available to a report. You can define a maximum of ten files per section of your report. You must define files before you insert fields.

DEFINE→VARIABLES



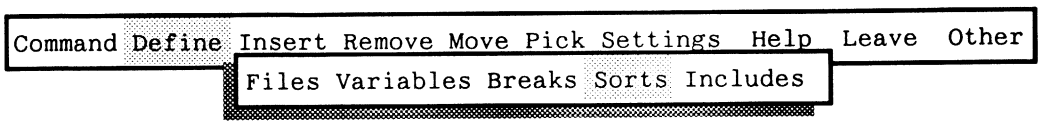
Use the DEFINE→VARIABLES command to define a variable for use in a report. Variables are typically used to perform calculations and display results from the report.

DEFINE→BREAKS



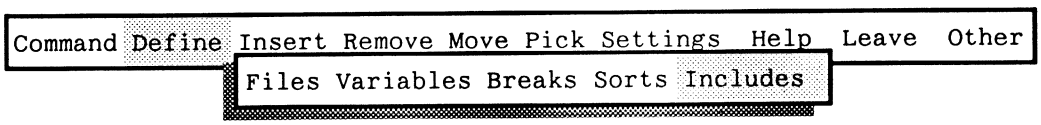
The DEFINE→BREAKS command lets you define a break group within the current report section. A break group lets you display fields or aggregated data whenever a value of a field within a file changes.

DEFINE→SORTS



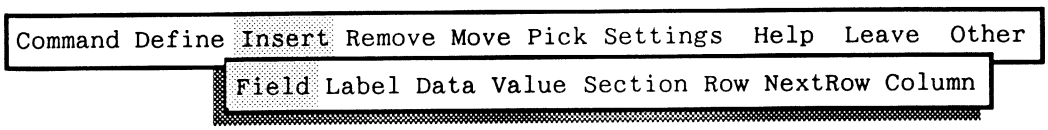
Before selecting the DEFINE→SORTS command, position your cursor in the report section in which you would like to sort data. You can sort your data alphabetically or numerically depending on the data type of the field by which you choose to sort. You can sort on any field in any file defined for the current report section or by a variable.

DEFINE→INCLUDES



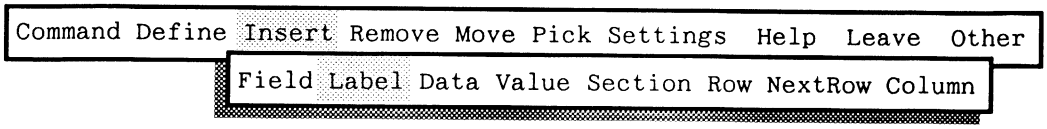
To define an include file in a report, select the DEFINE→INCLUDES command from the Report Writer horizontal menu. You can use include files to perform complex qualifications or lengthy calculations on report data. The Report Writer generates a PROGRESS FOR EACH statement for each section in your report. The field in which you enter the include file name determines whether the include file is inserted before the PROGRESS FOR EACH statement for the current section, after it, or within it. When you run your report using COMMAND→VIEW, the include file executes at the point at which it is defined.

INSERT→FIELD



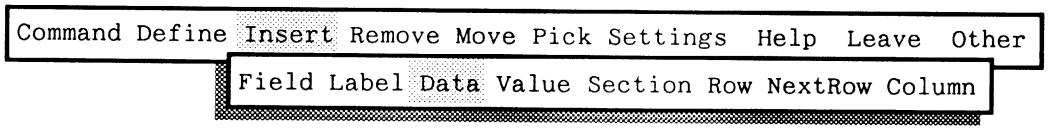
The INSERT→FIELD command lets you insert fields from a database file into a report section. You can only insert fields from files defined for the current section or those files defined for the parent section of the current section. Before selecting this command, position the cursor in the report section in which you want to insert fields. When you select this command, a Choices window appears. To insert a field from a file available to the current report section, you must first select a file from the files column, and then the fields from the fields column. Fields appear in the Report Writer edit screen in the order in which you insert them.

INSERT→LABEL



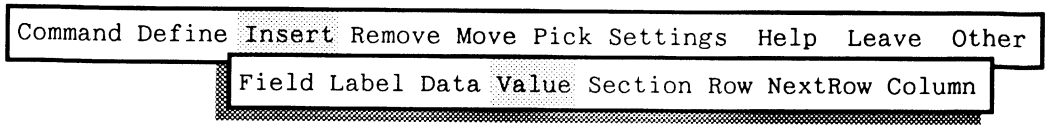
The INSERT→LABEL command lets you insert only the label of a field in a database file into the current report section. Before selecting this command, position your cursor at the point where you want the left most character of the first label to appear in the report. This command works exactly like the INSERT→FIELD command, The labels appear in the report in the order in which you insert them.

INSERT→DATA



The INSERT→DATA command lets you insert only the data area of a field in a database file into the current report section. Before selecting this command, position your cursor at the point where you want the left most character of the first data area to appear in the report. This command works exactly like the INSERT→FIELD command. The field data areas appear in the report in the order in which you select them.

INSERT→VALUE



The INSERT→VALUE command lets you insert a calculated value into your report at the position of your cursor. A Choices window containing valid choices for aggregates appears when you invoke INSERT→VALUE. Valid choices are:

`Page-number` — Inserts a page number at the current cursor location.

`Time` — Inserts the time of day at the current cursor location.

`Today` — Inserts the date at the current cursor location.

`Userid` — Inserts the user ID at the current cursor location.

When you enter an aggregate operation, an operation window appears. The following are valid functions for the operations window:

`Average` — Calculates the average of a set of data.

`Count` — Totals the number of entries for your current group.

`Maximum` — Returns the maximum value for a database field.

`Minimum` — Returns the minimum value for a database field.

`Total` — Returns the total value of the field for the entire database or break group. You can only use this function with numeric data types.

`Sub-average` — Returns the average of the field in a break group. You can only use this function with numeric data types.

`Sub-count` — Returns the total times the field value has been counted in a break group. You can only use this function with numeric data types.

`Sub-maximum` — Returns the maximum value of a field in a break group. You can only use this function with numeric data types.

`Sub-minimum` — Returns the minimum value of a field in a break group. You can only use this function with numeric data types.

`Sub-total` — Returns the subtotal of the field value in a break group. You can only use this function with numeric data types.

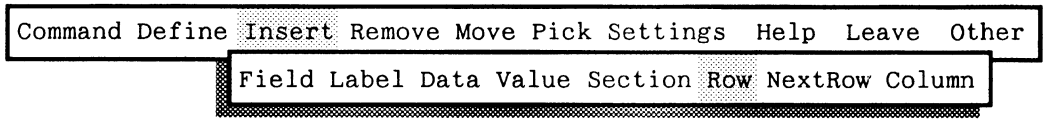
INSERT→SECTION

```
Command Define Insert Remove Move Pick Settings  Help  Leave  Other
        Field Label Data Value Section Row NextRow Column
```

The INSERT→SECTION command lets you create a new report section, which is equivalent to a new PROGRESS FOR EACH statement. When you select this command, the Structure of Report window appears. The window prompts you to enter a name for the section that your are inserting. Enter a section name of up to four characters in the Insert Section Name field. This entry appears in the margin of report areas when you return to the Report Writer edit screen. Enter a section title of up to 15 characters in the Section Title field. When you highlight this new section on the Report Writer edit screen, the entry in the Section Title field appears on the bottom line of the screen.

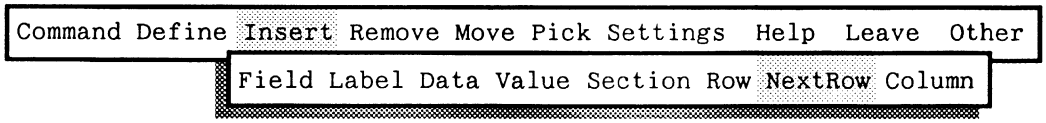
If you enter a new report section and save it to the report structure using the Report Structure option, the Input Files window appears automatically. Define files for the new section just as you would using the DEFINE→FILES command. You can define a maximum of ten files per report section. The new section appears in the margin of area and section names on the Report Writer edit screen.

INSERT→ROW



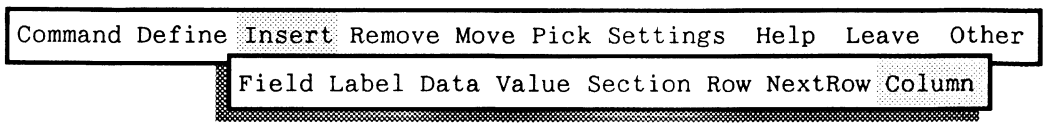
Use the INSERT→ROW command to insert a new row at the current cursor location. The current report row and all rows below it move down one row.

INSERT→NEXTROW



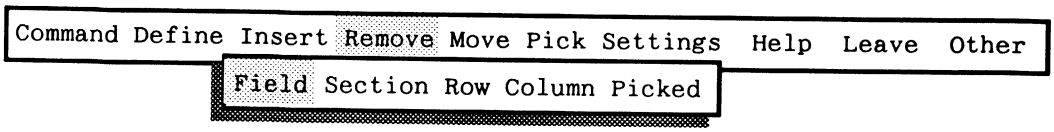
The INSERT→NEXTROW command lets you insert a new row immediately below the location of the cursor.

INSERT→COLUMN



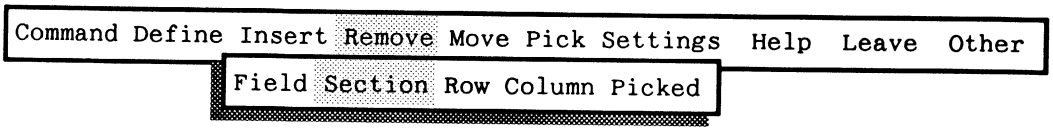
Use the INSERT→NEXTROW command to insert a column at the location of the cursor. You cannot insert a column if it would intersect a field, label, or data area.

REMOVE→FIELD



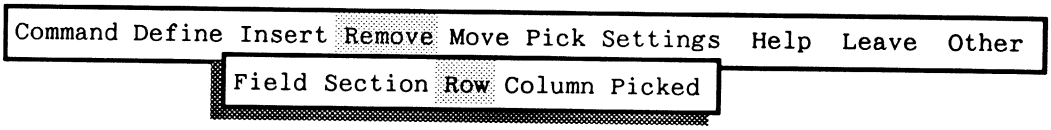
The REMOVE→FIELD command removes the field at the current cursor location. Position the cursor in the field that you want to delete and invoke REMOVE→FIELD to remove the entire field, including the label and data area.

REMOVE→SECTION



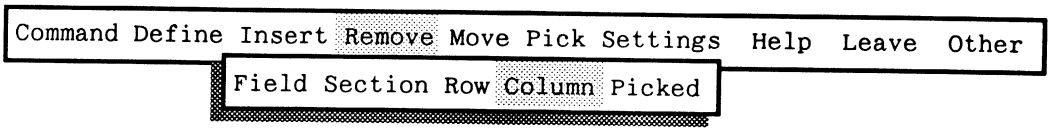
The REMOVE→SECTION command lets you remove any section from a report. When you select this command, the Structure of Report window appears. To delete a section from the current report, use the window to enter a section name into the Delete Section Name field. The Report Writer highlights the specified section and displays a confirmation prompt.

REMOVE→ROW



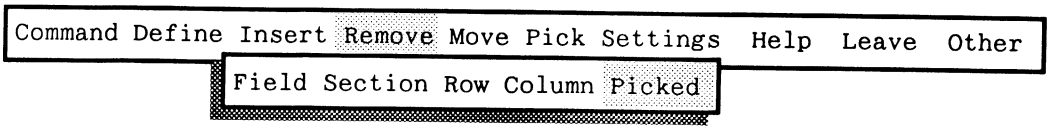
The REMOVE→ROW command removes a row and any data in it.

REMOVE→COLUMN



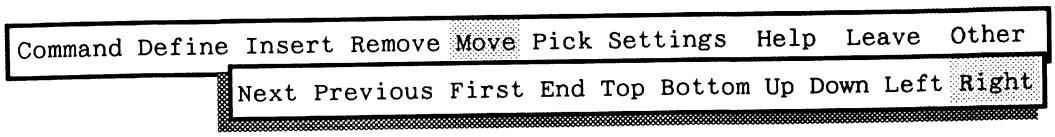
Use the REMOVE→COLUMN command to remove an empty report column.

REMOVE→PICKED



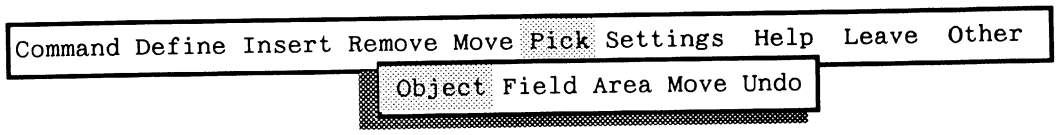
The MOVE→LEFT command scrolls 20 characters to the left when the report is wider than the screen.

MOVE→RIGHT



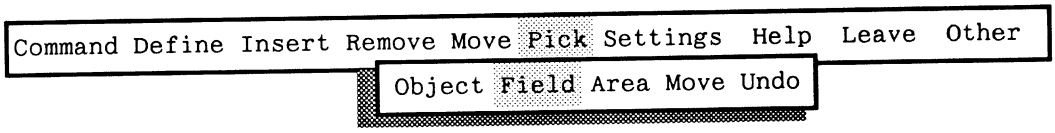
The MOVE→RIGHT command scrolls 20 characters to the right when the report is wider than the screen.

PICK→OBJECT



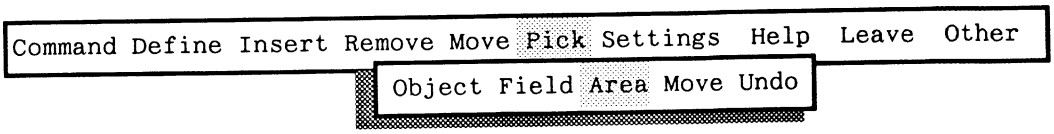
The PICK→OBJECT command highlights a label or a data area at the current cursor location. Locate the cursor in the area you want to pick. Then highlight the area and select REMOVE→PICKED to remove the object from the report, or PICK→MOVE to move the object elsewhere in the same report section.

PICK→FIELD



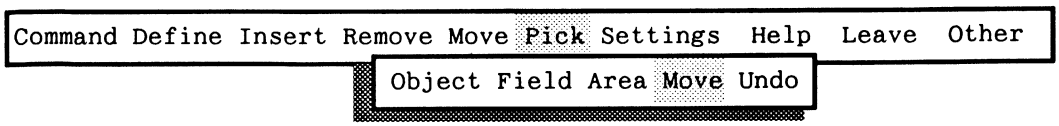
The PICK→FIELD command highlights the entire field at the current cursor location. Locate your cursor in the field you want to pick. Then highlight the field and select REMOVE→PICKED to remove the field from the report or PICK→MOVE to move the field elsewhere in the same report section.

PICK→AREA



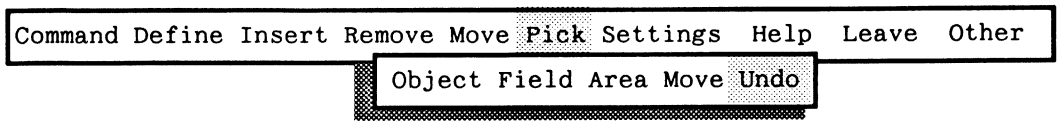
The PICK→AREA command selects a group of fields. Locate the cursor at any corner of the area you want to select. Invoke PICK→AREA and move the cursor to the opposite corner of the area you want to mark, then invoke PICK→AREA again to mark the area. (Note that you cannot mark an area that cuts through a field label or data area.) To delete the picked area, use REMOVE→PICKED. To move the picked area, use PICK→MOVE.

PICK→MOVE



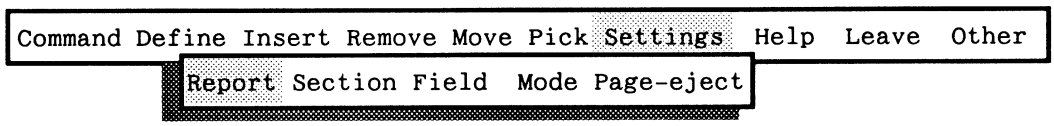
The PICK→MOVE command moves the currently picked field, object, or area to the current cursor location. Move the cursor to the position where you want the currently highlighted item to appear. Invoke PICK→MOVE and the upper- left corner of the item appears at the cursor position. You can only move a field or a data area to another report section if the database file for the field or data area is available for use in the new report section.

PICK→UNDO



The PICK→UNDO command deselects the currently highlighted field, object, or area.

SETTINGS→REPORT



When you select the SETTINGS→REPORT command, FAST TRACK displays the Report Settings window. You can update the following settings in the Report Settings window:

Report Name. The name of the report.

Title. Any string of alphabetic characters including spaces, up to 35 characters long. If you entered a report title when you created the current report, the title will appear in this field.

Report Width. A report can be up to 255 characters wide. This width is normally used for output to a printer. The default report width is 80 characters.

Page Size. This setting allows you to change the number of lines per page on your report. The default page size is 18 lines.

Output Device. This setting lets you set the output destination of the report output at run-time.

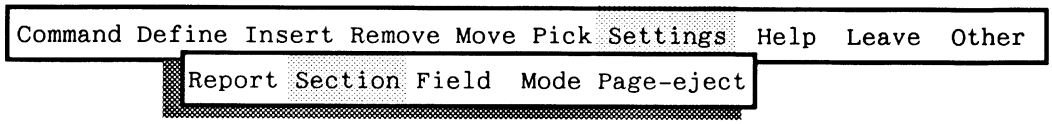
`Class/option`. Specify the output device further. You can specify an output class for the following output devices: **printer, spool, file**, and **include.**

`Sub-directory for Generated Procedures`. This is the name of the directory where you want FAST TRACK to store any FAST TRACK generated procedures.

`Can Be Run By`. This setting determines who can run the report. The default value for this setting is ? (unknown). The ? value lets you run the current report with FAST TRACK and PROGRESS versions of your current database.

`Report Description`. This setting is a description of your report. It is used for documentation purposes.

SETTINGS→SECTION



The `SETTINGS→SECTION` command lets you change the default setting for the current report section. Before selecting this command, position your cursor in the section that you would like to alter. When you select this command, the Section Settings window appears. The window lets you modify the following section settings:

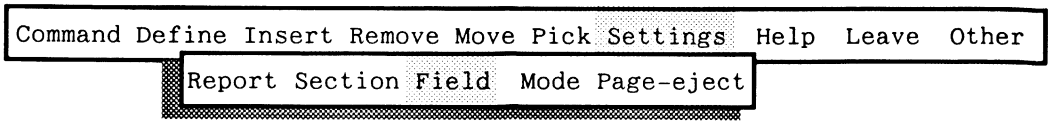
`Section Name`. Enter a new section name of up to 4 characters in length. This name appears in the margin containing report area and section names on the Report Writer edit screen.

`Title`. Enter a string of up to 15 characters in length. The section title appears at the bottom of the several screens in the Report Writer to let you know the section in which you are working.

`Label (Top/Side)`. Enter either **top** or **side**. This setting lets you specify whether you want to insert your data with top labels or side labels in the current section of your report. Once you insert a field into the current section, you cannot change this setting.

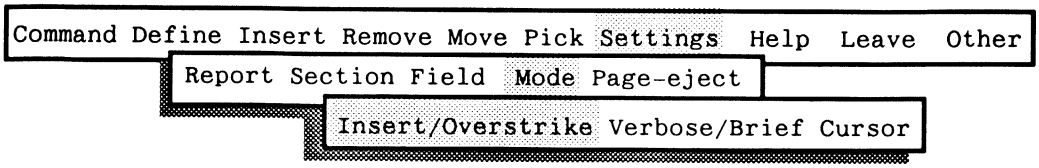
`Page-eject`. Enter **yes** or **no**. This setting lets you specify whether you want a page-eject after this section of your report.

SETTINGS→FIELD



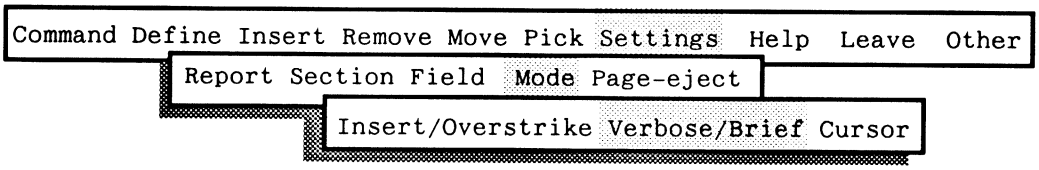
The SETTINGS→FIELD command lets you change the label and data format of a field in your report. Locate the cursor in the field for which you want to change the defaults, the select SETTINGS→FIELD and the Field Attribute Setting window appears. Now change the label and data format settings.

SETTINGS→MODE→INSERT/OVERSTRIKE



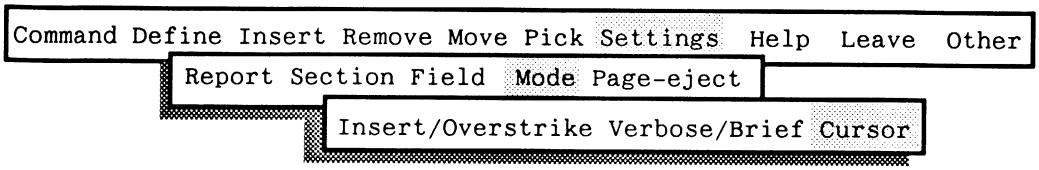
When you select the SETTINGS→MODE→INSERT/OVERSTRIKE command, FAST TRACK toggles between insert and overstrike mode. If you are in insert mode, FAST TRACK displays the mode setting in the lower right corner of your screen. In insert mode, an entered character is inserted at the current cursor location and the adjacent character moves one space to the right. In overstrike mode, the character you enter *replaces* the character at the current cursor location.

SETTINGS→MODE→VERBOSE/BRIEF



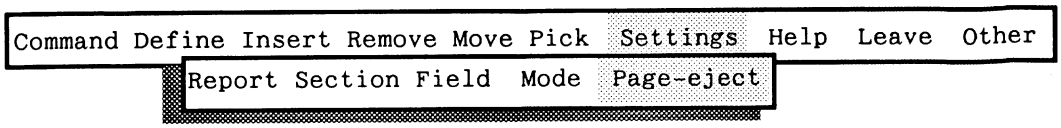
The SETTINGS→MODE→VERBOSE/BRIEF command lets you turn off the bar for section and area markers. You can still view the current section title in the middle of the status line at the bottom of your screen. To display the area and section markers, simply select this command again.

SETTINGS→MODE→CURSOR



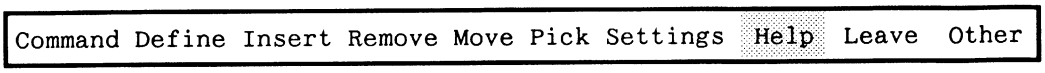
When you select the SETTINGS→MODE→CURSOR command, FAST TRACK displays the current cursor position on the bottom line of the screen.

SETTINGS→PAGE-EJECT



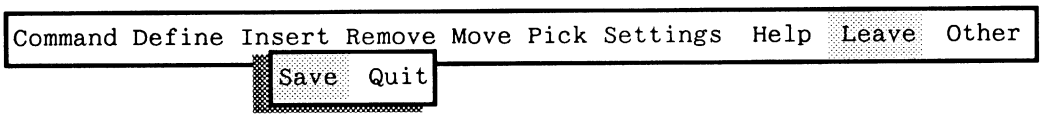
When you select the SETTINGS→PAGE-EJECT command, the Page Eject window displays the section names of all report sections. If you specify **No** (the default) for a particular section, a page eject does not occur after the section. If you specify **Yes**, a page-eject is inserted after the section.

HELP



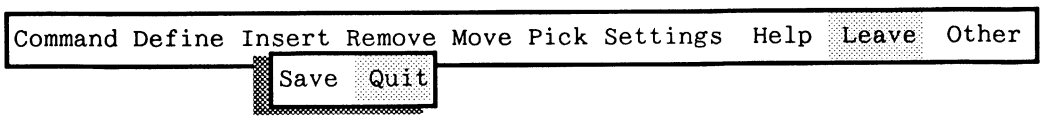
Use the HELP command to access help information on the Report Writer's horizontal menu commands.

LEAVE→SAVE



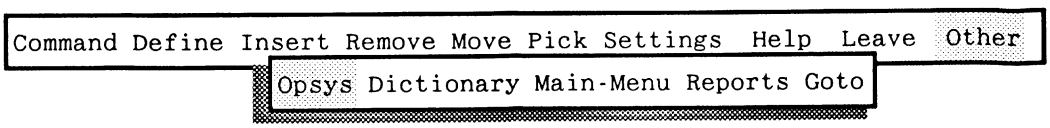
The LEAVE→SAVE command saves a report as an object in your current FAST TRACK database, and returns you to the FAST TRACK Main Menu.

LEAVE→QUIT



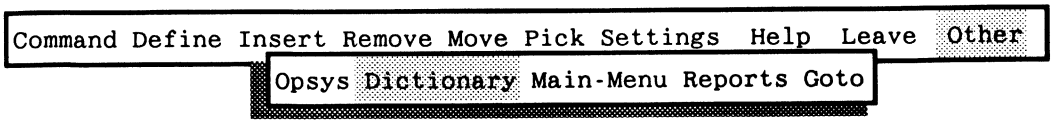
The LEAVE→QUIT command returns you to the FAST TRACK Main Menu without saving any changes in your current report.

OTHER→OPSYS



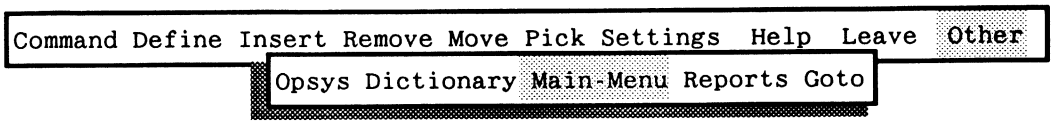
The OTHER→OPSYS command lets you temporarily leave FAST TRACK to use the operating system. You return to the application by typing **exit** from DOS, by pressing CTRL-D from UNIX, by typing **PROGRESS Exit** from BTOS/CTOS, or by typing **logout** from VMS.

OTHER→DICTIONARY



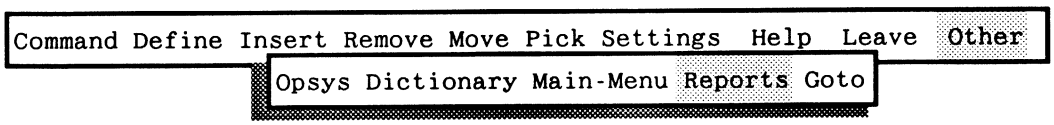
The OTHER→DICTIONARY command lets you use the PROGRESS Data Dictionary.

OTHER→MAIN-MENU



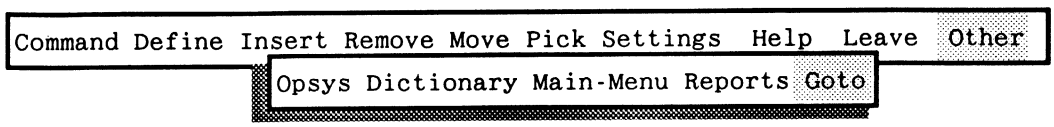
The OTHER→MAIN-MENU command returns control to the FAST TRACK Main Menu from any point in an editing session. FAST TRACK gives you a chance to save any changes before returning to the Main Menu.

OTHER→REPORTS



The OTHER→REPORTS command lets you access a variety of development reports containing information about all of the FAST TRACK objects in your application.

OTHER→GOTO



The OTHER→GOTO command lets you run FAST TRACK objects such as menus, QBFs, reports, or PROGRESS procedures. When you use this command the particular FAST TRACK object runs exactly as if it were being executed in an application.

Appendix D
FAST TRACK File Types

This appendix describes the file types for each FAST TRACK module. An understanding of FAST TRACK file types should help you avoid creating duplicate file names. For example, if you define a QBF with the name **cust**, FAST TRACK generates PROGRESS files `cust.p`, `custa.p`, `custb.p`, among others, so you should not give anything else in the system any of these names.

In general, the Menu Editor, Report Writer, and QBF Generator produce files that can be used in either FAST TRACK or PROGRESS. The Screen Painter, however, produces a unique file type — form files, which are denoted by a `.f` extension — that can only be referenced by a PROGRESS procedure. Table D–1 summarizes the file types for all the FAST TRACK modules.

Table D–1: FAST TRACK File Types

Module	File Types
Menu Editor	procedure (`.p`), compiled (`.r`)
Screen Painter	form (`.f`)
Report Writer	procedure (`.p`), compiled (`.r`), include (`.i`)
QBF Generator	procedure (`.p`), compiled (`.r`), include (`.i`)

NOTE: When creating files, do not use the following filenames: REPORT, MENU, FT, FORM, MAINT, HELP, or QBF. These are names of FAST TRACK subdirectories. If you inadvertantly give a file one of these names, FAST TRACK will not operate properly.

D.1 MENU EDITOR FILES

In the Menu Editor, FAST TRACK generates a procedure (.p) file when you execute the COMMAND→GENERATE command. You provide the name for this file, and FAST TRACK automatically appends the .p extension if you do not supply it yourself. At the same time, FAST TRACK creates a compiled version of the file, which is denoted by a .r extension.

For a typical application, the procedure file associated with a top-level menu serves as the start-up procedure for the application. In certain cases, however, you might want to run the procedure from the PROGRESS editor. This is a perfectly valid technique, but use it sparingly because FAST TRACK itself can likely achieve the same result.

Normally a submenu is tied to a choice of another menu, specified in that menu's choice type as menu. Instead of using this standard structure to invoke a submenu, you can call the submenu as a procedure by naming the submenu's .p file. To do so, you use the Menu Choice Settings window shown in Figure D-1.

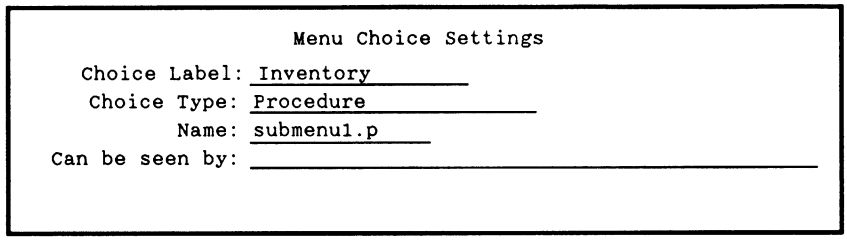


Figure D-1: Menu Choice Settings

In this example, the Choice Type of Procedure tells FAST TRACK to execute a procedure file. When you invoke the menu choice, FAST TRACK runs the procedure (submenu1.p) specified in the Name field.

D.2 SCREEN PAINTER FILES

In the Screen Painter, FAST TRACK generates a form (.f) file when you execute the COMMAND→GENERATE command. The full name of the file consists of the the form name, which you specify in the Screen Painter initialization window, and the .f extension, which FAST TRACK automatically appends.

A form file contains the PROGRESS form definition for the screens created in the Screen Painter. You can use a form file by including it in a PROGRESS procedure. For example, the following PROGRESS code executes an UPDATE statement on the data specified in the custf.f form:

```
FOR EACH customer:
      UPDATE {custf.f}.
```

Here, ensure that you enclose the form name in braces. If you want, you can also use the PROGRESS SET, DISPLAY, and PROMPT-FOR statements with a form file. If you use the PROMPT-FOR statement, however, you must also use the ASSIGN statement; otherwise, FAST TRACK displays a warning message.

NOTE: FAST TRACK itself does not reference form files created with the Screen Painter. If you do not intend to use a form file with a PROGRESS procedure, you do not need to use COMMAND→GENERATE in the Screen Painter.

D.3 REPORT WRITER FILES

In the Report Writer, FAST TRACK generates a procedure (.p) file when you execute the COMMAND→GENERATE command. As with the Menu Editor and QBF Generator, a compiled (.r) file is generated along with the .p file. Unlike other procedure files, however, a Report Writer procedure file is actually a driver program for a user-generated report.The purpose of a Report Writer procedure file is two-fold:

- It maintains a report's output device information.
- It invokes an include file that contains the PROGRESS code for the report.

You can execute a Report Writer procedure file from either the PROGRESS editor or through a FAST TRACK menu. If you choose the first method, you cannot change the output device for the associated report using the "Update Output Devices" option on the Maintenance menu. If you want the end-user to have this flexibility, tie the report to a menu choice and specify Report in the Choice Type field of the Menu Choice Settings window. Then specify the name of the report in the Name field of the window.

For each Report Writer procedure file, FAST TRACK generates a secondary procedure file that contains the actual PROGRESS code for the report. FAST TRACK automatically appends the letter r to this procedure's filename (not the file extension). Additionally, if the name of the report is less than seven characters, FAST TRACK pads it with the underscore character. For example, if your original report name is x, FAST TRACK converts it to the following:

```
x_____r.p.
```

When you execute a primary procedure file in the Report Writer, the file automatically executes a PROGRESS RUN statement. The parameter to the RUN statement is the name of the secondary procedure file. For example:

```
RUN  x_____r.p.
```

If you want, you can run a secondary procedure file from the PROGRESS editor. If you use this method, note that the report output is always directed to the terminal.

The Report Writer also generates include (.i) files. Up to nine files, one corresponding to each section in a report, can be generated. Each include file contains the header and form definitions corresponding to a section.

FAST TRACK always appends an integer to the end of an include filename (not the file extension). Thus, if the report name is rep, the associated include files are:

```
rep1.i  rep2.i  rep9.i...
```

D.4 QBF FILES

In the QBF Generator, FAST TRACK automatically generates a procedure (.p) file when you create a QBF. The name you assign to the QBF is appended with a .p extension. For example, if you have given your QBF the name **custq**, FAST TRACK names the procedure `custq.p`. At the same time, FAST TRACK creates a compiled version of the file, which is denoted by a .r extension.

As with a menu, or a report, a QBF procedure can be run from the PROGRESS editor, or tied to a menu choice. To tie a QBF to a menu choice, specify QBF in the `Choice Type` field in the Menu Choice Settings window, then specify the QBF object name, and run run the QBF procedure file as shown in Figure D–2.

```
                    Menu Choice Settings
   Choice Label: Inventory
    Choice Type: Procedure
           Name: someqbf.p
Can be seen by:
```

Figure D–2: Menu Choice Settings

In this example, the `Choice Type` of Procedure tells FAST TRACK to execute the procedure (`someqbf.p`) specified in the `Name` field.

In addition to the a QBF's main procedure file and corresponding compiled file, the QBF Generator creates two sets of additional files. Each set consists of a procedure file, a compiled (.r) file, and an include (.i) file. The first set (denoted by an a preceding the dot character) handles the code for standard QBF processing. The second set (denoted by a b preceding the dot character) handles the code for using a QBF with the `JOIN` command. To avoid duplicate filenames, never use a or b as a suffix when you name a file.

In all, for a QBF named *someqbf*, the QBF Generator produces the following files:

someqbf.p someqbf.r	Main QBF Files
someqbfa.p someqbfa.r someqbfa.i	Standard QBF Code
someqbfb.p someqbfb.r	QBF JOIN Code

D.5 FAST TRACK DATA FILES

All FAST TRACK objects (menus, screens, reports, and QBFs) are maintained in the FAST TRACK database. When you invoke the `Dump FT Data Files` option from the Maintenance menu, all FAST TRACK objects are dumped to data files.

FAST TRACK data files are distinguished from other files by their underscore prefix and a `.d` extension. FAST TRACK creates the following files when you invoke `Dump FT Data Files`:

```
_choice.d
_fdef.d
_form.d
_kb.d
_menu.d
_outdev.d
_prog.d
_qbf.d
_ragg.d
_repexp.d
_report.d
_rgroup.d
_rlevel.d
_row.d
_rqual.d
_usrdev.d
```

NOTE: Although PROGRESS and FAST TRACK maintain data as objects in their respective databases, only FAST TRACK actually dumps files beginning with the underscore character. Refer to the *Programming Handbook* for additional information on PROGRESS data files.

D.6 SCRATCH FILES

FAST TRACK also creates various scratch files that it uses to temporarily store data. These files are automatically deleted, but if your system crashes, they could remain on your disk. Scratch files are denoted by either the prefix `erase` as in `eraseme.p`, or the extension `.tmp` as in *somefile*`.tmp`. When you discover a scratch file, feel free to delete it if you know no one else is using the database.

Appendix E
Keyboard Settings

This appendix describes how to change the default key settings on UNIX and VMS systems. You should only use the procedures outlined in this appendix when it is absolutely necessary to change the default key settings. For example, if one or more keys on your terminal are dedicated to a hardware related function, this would be adequate justification for changing the default key settings.

In this manual and in the *PROGRESS FAST TRACK Tutorial*, keys and key sequences are denoted by both a key label such as F1 and a name label such as OPTIONS.

E.1 CHANGING KEY ASSIGNMENTS

In order to change key assignments, you need to create a custom `protermcap` file if you are an individual user; or back up and modify the existing system `protermcap` file if you are the system administrator.

The `protermcap` file is located in the directory that contains the system files for FAST TRACK and PROGRESS. If you used the default directory structure when you installed FAST TRACK and PROGRESS, `protermcap` is in the `dlc` directory.

Because FAST TRACK runs on various terminals, `protermcap` defines keyboard layouts for more than 50 terminals. Before editing `protermcap`, ensure that you have a backup copy of the file. Consult your system administrator if necessary.

The best way to create a custom copy of `protermcap` is to copy only that part of the file that pertains to your terminal. To do this, load `protermcap` into an ASCII text editor, scroll through the file until you find your terminal definition, then copy the definition to a new file. If you are an individual user, copy the file to your working directory and name it `protermcap`. If you are the system administrator, copy the file to the system directory.

The following steps summarize the process the individual user should use to create a custom `protermcap` file.

1. Examine FAST TRACK default key settings. When you make changes to protermcap, avoid re-defining a key or key sequence to an existing definition. Default settings are listed in Table E-2.

2. Copy the appropriate terminal definition from the original `protermcap` file to a new file called `protermcap` in your working directory.

3. Modify the keyboard settings in your new copy of `protermcap` to those you want to implement. See section E-3 for information on how to modify settings for keys and key sequences.

4. Save your edited copy of the custom `protermcap` file.

5. Define a PROTERMCAP variable for your operating system (see section E.4).

6. Load FAST TRACK and confirm that your new keyboard settings have taken effect. Do this by selecting `Settings` from the Main Menu and examining the FAST TRACK Settings window (see Chapter 2).

E.2 DEFAULT KEY SETTINGS

In addition to the label keys used by FAST TRACK, you can also define other label keys available in PROGRESS. Table E-1 lists these keys as well as the standard CTRL keys and function keys for the various FAST TRACK and PROGRESS products.

Table E-1: PROGRESS and FAST TRACK Keys

Key Function	Editor	Procedure Execution	Standard Keyboard Key	Standard Control Key			Standard Function Key			Allowed in ON
				DOS OS/2	UNIX VMS	BTOS/ CTOS	DOS OS/2	UNIX VMS	BTOS/ CTOS	
ABORT	✓	✓		Ctrl-Alt -Del	**	ACTION/				
APPEND-LINE	✓			Ctrl-A	Ctrl-A	CODE-A	ALT-F2 F11 (OS2)	F12	SHIFT-F12	
BACK SPACE	✓	✓	BACK SPACE							✓
BACK-TAB	✓		SHIFT-TAB (DOS & OS/2)	Ctrl-U	Ctrl-U	CODE-U			CODE-TAB	✓
BELL										✓
BLOCK	✓			Ctrl-V	Ctrl-V	CODE-V	ALT-F4	F14	SHIFT-F4	
* BOTTOM-COLUMN						CODE- ↓	ALT-B	ESC, ↓ ESC,B		✓
BREAK-LINE	✓			Ctrl-B	Ctrl-B	CODE-B	ALT-F1 F11 (OS2)	F11	SHIFT-F1	
BTOS/CTOS-END		✓	EXIT EXEC(BTOS) or FINISH EXEC(CTOS) PROGRESS EXIT (PROGRESS Utility to quit CONTEXT with PAUSE.)							
* CANCEL-PICK							ALT-X	ESC,X	CODE-CANCEL	✓
* CHOICES						CODE-C	ALT-C	ESC,C		✓
CLEAR	✓	✓		Ctrl-Z	Ctrl-Z	CODE-Z	F8	F8	F8	✓
CURSOR-UP	✓	✓	↑	Ctrl-K	Ctrl-K					✓
CURSOR-DOWN	✓	✓	↓	Ctrl-J	Ctrl-J					✓
CURSOR-LEFT	✓	✓	←							✓
CURSOR-RIGHT	✓	✓	→	Ctrl-L	Ctrl-L					✓
DELETE-CHARAC-TER	✓	✓	DEL DELETE [1]							✓
* DELETE-COLUMN						CODE-SHIFT-D	ALT-Z	ESC,Z		✓
* DELETE-FIELD						CODE-J	ALT-D	ESC,D		✓

(continued on next page)

* FAST TRACK interface keys

** UNIX: \ (Depends on UNIX stty setting for quit)
VMS: Ctrl-Y, STOP

Table E-1: PROGRESS and FAST TRACK Keys (continued)

Key Function	Editor	Procedure Execution	Standard Keyboard Key	Standard Control Key			Standard Function Key			Allowed in ON
				DOS OS/2	UNIX VMS	BTOS/ CTOS	DOS OS/2	UNIX VMS	BTOS/ CTOS	
DELETE-LINE	✓			Ctrl-D	Ctrl-D	CODE-D	F10	F10	F10	
DOS-END		✓		Type EXIT						
ENDKEY		✓								✓
END-ERROR	✓	✓	ESC (DOS) [1 2] CANCEL or FINISH	Ctrl-E	Ctrl-E	CODE-E	F4	F4	F4	✓
ERROR		✓								✓
FIND	✓			Ctrl-F	Ctrl-F	CODE-F	ALT-F3	F13	SHIFT-F3	
GET	✓			Ctrl-G	Ctrl-G	CODE-G	F5	F5	F5	
GO	✓	✓		Ctrl-X	Ctrl-X	CODE-X	F1	F1	F1	✓
GOTO LINE *					ESC-G	CODE-SHIFT-G	ALT-G	ESC,G		✓
HELP	✓	✓ ***	HELP	Ctrl-W	Ctrl-W	CODE-W	F2	F2	F2	✓
HOME	✓	✓	HOME		ESC-H				CODE-NEXT -PAGE	✓
INSERT-COLUMN *						CODE-SHIFT-C	ALT-N	ESC,N		✓
INSERT-FIELD *						CODE-I	ALT-I	ESC,I		✓
INSERT-FIELD-DATA *						CODE-SHIFT-L	ALT-F	ESC,F		✓
INSERT-FIELD-LABEL *							ALT-E	ESC,E		✓
INSERT-MODE	✓	✓	INSERT OVERTYPE [1]	Ctrl-T	Ctrl-T	CODE-T	F3	F3	F3	✓
LEFT-END	✓		CTRL-← (DOS & OS/2)						CODE-←	
MAIN-MENU *						CODE-M	ALT-M	ESC,M		✓
MOVE *						MOVE	ALT-V	ESC,V		✓
NEW-LINE	✓			Ctrl-N	Ctrl-N	CODE-N	F9	F9	F9	
OS/2 END		✓		Type EXIT						
PAGE-DOWN	✓		NEXT PAGE or PG DN				ALT-F6	16	SHIFT-F6	
PAGE-UP	✓		PREV PAGE or PG UP				ALT-F5	15	SHIFT-F5	
PICK *						CODE-K	ALT-P	ESC,P		✓

* FAST TRACK interface keys

*** Allowed only if help procedure, applhelp.p, exists

(continued on next page)

1. BTOS/CTOS
2. OS/2

Table E-1: PROGRESS and FAST TRACK Keys (continued)

Key Function	Editor	Procedure Execution	Standard Keyboard Key	Standard Control Key			Standard Function Key			Allowed in ON
				DOS OS/2	UNIX VMS	BTOS/ CTOS	DOS OS/2	UNIX VMS	BTOS CTOS	
PICK-AREA *						CODE-SHIFT-A	ALT-W	ESC,W		✓
PICK-LABEL-DATA *						CODE-Q	ALT-Q	ESC,Q		✓
PUT	✓			Ctrl-P	Ctrl-P	CODE-P	F6	F6	F6	
RECALL	✓	✓		Ctrl-R	Ctrl-R	CODE-R	F7	F7	F7	✓
REPAINT	✓						ALT-P	ESC,P		✓
REPORTS *						CODE-SHIFT-T	ALT-A	ESC,A		✓
RESUME DISPLAY	✓	✓		Ctrl-Q	Ctrl-Q					
RETURN	✓	✓	RETURN or NEXT	Ctrl-M	Ctrl-M					✓
RIGHT-END	✓		CTRL-→ (DOS & OS/2)	Ctrl-E	Ctrl-E		F4	F4	CODE-→	
SCROLL-LEFT *							ALT-L	ESC,L	SHIFT-←	✓
SCROLL-RIGHT *							ALT-R	ESC,R	SHIFT-→	✓
SEARCH	✓					CODE-F	ALT-F	ESC,F		
SETTINGS *						CODE-S	ALT-S	ESC,S		✓
STOP		✓		BREAK	**** Ctrl-C	ACTION-CANCEL				✓
STOP-DISPLAY	✓	✓		Ctrl-S	Ctrl-S					
TAB	✓	✓	TAB	Ctrl-I	Ctrl-I					✓
TOP-COLUMN *							ALT-T	ESC, CURSOR UP ESC,T		✓
UNIX-END		✓			***** Ctrl-D					
VMS-END		✓			Type logout					

* FAST TRACK interface keys
**** Depends on the UNIX stty setting for intr
***** Depends on the UNIX stty setting for eof

E.3 REDEFINING KEYS

To change a key label definition in `protermcap`, all you need to do is replace the label's existing key or key sequence definition. The key or key sequence definition appears in parenthesis in the `protercamp` file, directly after the key label name. For example, the OPTIONS key is defined as the following:

```
OPTIONS(CTRL-O)=^O:\
```

Additional terminal specific information appears after the closing parenthesis in the `protermcap` file. You must refer to your terminal user's manual for information on the codes used by your terminal.

To change a key label's definition, just change the key or key sequence inside the parenthesis, then append the terminal code. For example, to change [OPTIONS] to [CTRL]-[U], make the following change:

```
OPTIONS(CTRL-U)=^U:\
```

NOTE: If you have trouble with a `protermcap` entry, first check to be certain that every line (except the last one) ends with a backslash. Also be sure you have not entered a space after the backslash.

E.4 DEFINING THE PROTERMCAP VARIABLE

After you create a `protermcap` file and store it in your working directory, you must set the protermcap variable to the name of your alternate `protermcap` file. If you are working on a UNIX machine, include the following line in your `.profile` file, or enter it from the command prompt.

```
PROTERMCAP=protermcap;export PROTERMCAP
```

Include this line for VMS systems:

```
DEFINE PROTERMCAP PROTERMCAP.DAT
```

Now, run FAST TRACK. If you left any required entries out of the `protermcap` file, FAST TRACK prints an error message and you can make corrections and try again. Otherwise, call up the Settings option from the Main Menu to ensure that your new key definitions have taken effect.

To go back to using the default protermcap during a UNIX session in which you explicitly defined protermcap, use the UNIX command:

```
unset PROTERMCAP
```

If unset is not available on your machine, you may need to logout and enter UNIX again.

On VMS, use this command:

```
DEASSIGN PROTERMCAP
```

If you decide that you want to make modifications to the system `protermcap` file, you must add those modifications to the `protermcap` file in the directory where you installed FAST TRACK. Before making modifications, make a copy of thc original `protermcap` file.

Symbols

Numbers

A

B

C

D

E

F

G

H

I

K

L

M

N

O

P

Q

R

S

T

V

W