
April 1996

**QPL Reference
Manual
Version 4.0**

Questionnaire Programming Language
Version 4.0
April 1996

The QPL software and manuals were written by Kevin Dooley, General Government Division, U.S. General Accounting Office. (Email: dooleyk.ggd@gao.gov)

The U.S. General Accounting Office holds no copyright on this manual or the QPL software. The public may obtain this software and the manuals from GAO's World Wide Web page at <http://www.gao.gov>.

askSam is a trademark of Seaside Software, Inc.
dBase III+ and **dBase IV** are trademarks of Ashton-Tate, Inc.
Hercules is a trademark of Hercules Computer Technology, Inc.
Lotus 1-2-3 and **Lotus Symphony** are trademarks of Lotus Development Corp.
MS-DOS and **QBasic** are trademarks of Microsoft Corp.
NetWare is a trademark of Novell, Inc.
PC-DOS is a trademark of IBM Corp.
SAS/PC is a trademark of SAS Institute, Inc.
SPSS/PC+, **SPSS for Windows**, and **SPSS-x** are trademarks of SPSS Inc.
TAWK is a trademark of Thompson Automation, Inc.
WordPerfect is trademark of WordPerfect Corp.

Preface

The Questionnaire Programming Language (QPL) software provides a way of quickly and reliably automating survey data. This software enables you to quickly write computer-aided telephone interview (CATI) and data-entry programs that make it easy for interviewers to correctly capture information from respondents or files. It also provides a way of reliably editing the data that have been collected and quickly preparing them for analysis with statistical or text-retrieval software.

This software has been developed and revised by GAO since 1986. This fifth revision of the software reflects many things that we have learned using it. We have found that using QPL software is most appropriate in situations where survey or data-entry tasks need to be completed very quickly. It is also appropriate when information is being summarized on structured data collection forms. If evaluators must be used to fill out these forms, it is faster to allow them to enter information directly into the computer using this software than to have them write the information on paper and then send the forms to a keypuncher.

The QPL software attempts to automate as much of the survey process, short of actually writing and asking survey questions, as possible. This provides a number of benefits:

- Survey designers do not also have to be computer programmers. The QPL language can be learned in about one hour by people who are experienced at writing questionnaires.
- There is virtually no programming time required after a questionnaire has been written. It is actually easier for a questionnaire designer to write the questionnaire in the QPL language than to draft it by hand or using a word processor. A QPL utility program will automatically create a standard-looking questionnaire form that evaluators may review and comment on. Thus, the moment you have finished pretesting an instrument, you have a complete CATI or data-entry program that is ready to be used.
- Training staff to use the software is easy. The CATI and data-entry software use “user-friendly” software design, which make it easy to learn. Generally, it takes only about 15 minutes to train people to use the software. This leaves more time to train them on the subject of the survey itself and how to speak with respondents on the telephone.
- This version of the QPL software allows you to use networked PCs to simplify the administration of a telephone survey. With a network, all of your interviewers can simultaneously use the same list of calls and the same data file for the results.

- The data can be edited and reviewed quickly. The CATI and data-entry programs prevent many errors that could be made by interviewers. Skip instructions and range limitations are rigidly enforced so you do not need to check the data for invalid answers. If errors are discovered, it is easy to correct the raw data file. For example, you only have to tell the data editing program what record needs to be changed, and then change a “Yes” to a “No.” The software worries about what internal codes need to be changed and where they are in the data file.
- Statistical analysis can begin before all the data are collected. The system automatically generates SPSS and SAS programs that read and document the data file. Thus, you can begin analyzing the data as soon as they arrive. You only need to add the particular statistical procedures you want run. The data file itself can also be translated directly into a Lotus spreadsheet or dBase data file.
- Responses to open-ended questions can be easily entered and analyzed. An improvement in this version of the software allows virtually unlimited amounts of text to be entered in response to open-ended questions. In addition to being able to collect this type of information within a structured format, it can automatically create a text-based data file that can be analyzed with the askSam text-retrieval and hypertext software system.

This manual is a reference to the QPL language, and to the programs that are used to prepare the data for analysis. It is intended to be used by the person who develops and analyzes the survey. Separate manuals are provided for interviewers that contain instructions on how to use the CATI and data-entry programs. Instructions on installing this software on your computer are given in appendix I.

Contents

Preface		i
Chapter 1		1
System Description		2
	COLLECT Program	2
	NCOLLECT Program	2
	MONITOR Program	2
	KEYPUNCH Program	2
	CONVERT Program	3
Chapter 2		4
Language Introduction		5
	Program Lines	5
	Command Lines	6
	Comment Lines	7
	Blank Lines	7
	Question and Answer Text Lines	8
Chapter 3		10
Types of Questions		10
	Multiple Choice	10
	Check-All-That-Apply	18
	Number	23
	Short Answer	32
	Open-Ended Answer	39
	Date	43
	Current Date	51
	Current Time	53
	Version Number	58
	Void	60
	Random Number	61

Chapter 4		65
Skip Instructions	Unconditional Skips	65
	Conditional Skips	68
	Equation Processor	72
	Operators	72
	Variables	73
	Keywords	74
	Functions	78
	Exiting the Questionnaire	84
<hr/>		
Chapter 5		88
Data File Format	Fixed Format Files	88
	Missing Values	91
	Answer Field Sizes	92
<hr/>		
Chapter 6		98
Scheduling Interviews	Automated Call Sheets	99
	Creating and Using Control Files	100
	Merging the Control and Questionnaire Programs	103
	Additional Control File Commands	110
	Importing Control Data Files	118
<hr/>		
Chapter 7		119
Foreign Languages	Reconfiguring Your Computer	119
	Translating QPL Error Messages	121
<hr/>		
Chapter 8		124
Running the	Multiple Source Files	126
COMPILE Program	Options	128
	Errors	132
	Stopping the COMPILE Program	134

Chapter 9		135
Running the COLLECT Program	Options	137
	Screen Display	142
	Setting Screen Colors	143
	Organizing the Interviewer's Programs	147
<hr/>		
Chapter 10		150
Running the NCOLLECT and MONITOR Programs	Novell NetWare	150
	Starting the NCOLLECT Program	151
	Files Created by NCOLLECT	152
	Tracking Interviews	153
	Options	154
	Network Setup Issues	158
	Trouble Shooting	159
	Using the MONITOR Program	163
	Starting the MONITOR Program	164
	Options	165
	Monitor Menu	166
	Print Menu	167
	Quit	168
<hr/>		
Chapter 11		169
Running the KEYPUNCH Program	Options	171
	Screen Colors	173
	Titles and Subtitles	173
	Organizing the Interviewer's Programs	174
	Using askSam Text-Based Data Files	177
	Creating an askSam File	177
<hr/>		
Chapter 12		180
Running the CONVERT Program	Options	182
	CONVERT Menu	183
	Using the Menu	184
	Variables Menu	186
	Records Menu	188
	Select by Numeric Search	189
	Select by String Search	190
	Options Menu	191

Data Menu	192
askSam Import Files	193
dBase III+ and IV Files	195
Copy Data File	195
Translate Data File	196
Comma and TabDelimited Files	198
Lotus Files	201
Program Menu	203
Thompson Awk Programs	204
Microsoft QBasic Programs	206
SAS Programs	208
SPSS Programs	215
Formatted Question List	220
Titles, Subtitles, and Page Breaks	223
Cross-Reference List	225
File Menu	228
View Text File	229
Print Text File	229
Erase File	232
Rename File	232

Appendixes

I. Installing the QPL Programs	234
II. Changes From Previous Versions	238
III. Language Summary	245
IV. Command Line Options	258
V. File Name Extensions	261

Index

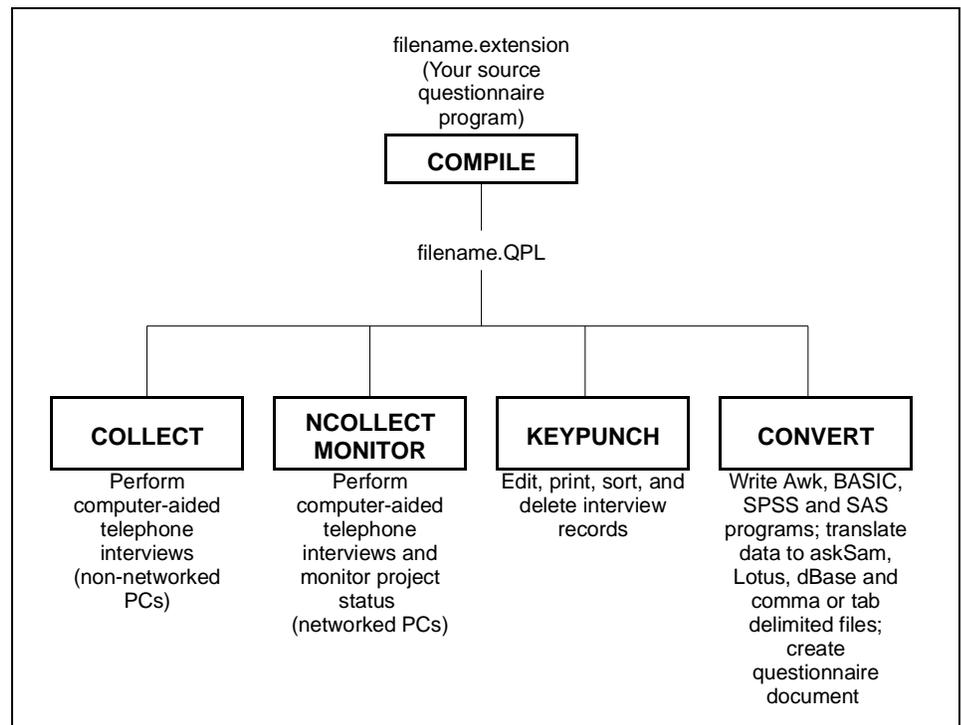
264

System Description

The Questionnaire Programming Language consists of a set of IBM/PC programs that you can use to automate many of the activities involved in gathering and preparing survey data for analysis. Using this software, you can write complex computer-aided telephone interview or data-entry programs that are easy to use and provide a high degree of control over what information may be entered. Interviewers can be trained to use the CATI software in only minutes, and completed interview records can be edited quickly and accurately. Once you have written your questionnaire program, you can use other QPL system programs to automatically generate formatted questionnaire documents, SPSS or SAS analysis programs, Lotus or dBase data files, or askSam text-based data files.

To use these programs, however, you must first write your questionnaire in the QPL language using a word processing program. After you have saved your questionnaire as an ASCII text file, you must compile it into a shortened form using the COMPILE program. (See chapter 8.)

QPL System of Programs



COLLECT Program

The COLLECT program uses your questionnaire to generate a CATI program. It reads your compiled questionnaire and displays each question on the computer screen, one at a time, and then waits for the interviewer to type a response. The interviewer can page forward and backward through the questionnaire to make corrections or review answers. The record of the interview is added to the data file after the interviewer answers the last question. The COLLECT program has many editing features that simplify the interviewer's job and improve the accuracy of the data being entered. (See chapter 9 and *QPL Data Collection Program—Version 4.0.*)

NCOLLECT Program

The NCOLLECT program is the network version of the COLLECT program. In addition to doing everything the COLLECT program does, it also has a number of features that allow you to let multiple interviewers simultaneously share the same data file. This capability can greatly simplify the administration of a telephone survey: All of the interviewers can share the same list of outstanding calls (called a control list) and record their results in the same data file. It also automatically keeps a record of when each interviewer accessed the system. (See chapter 10.)

MONITOR Program

The MONITOR program is used to track the progress of all the interviewers as they use the system on a network. This program will display when an interviewer accessed the system, what case he or she is working on at the moment, what cases they have worked on, whether they saved an interview record, and how many records have been added to the data file. (See chapter 10.)

KEYPUNCH Program

The KEYPUNCH program may be used to edit records of telephone interviews that were obtained with the COLLECT program, or to enter new records. Like the COLLECT program, this program has many editing features that make it easy to enter and change data accurately. In addition, it can print records in a format that makes them easy to read and verify against other documents. It also can sort the interview records and create an askSam-compatible text-based data file. (See chapter 11 and *QPL Data Editing Program—Version 4.0.*)

CONVERT Program

The CONVERT program performs a variety of functions to prepare your questionnaire data for analysis. These include

- generating Awk, QBasic, SAS, and SPSS statistical analysis programs that can process a QPL data file,
- translating a QPL data file to other formats, including askSam, Lotus, dBase, and tab and comma delimited formats,
- translating your QPL data file from the record layout of one version of your questionnaire to the record layout a newer version,
- copying selected records from one QPL data file to another,
- translating the text that was entered into open-ended questions to a generic word-processor file (which may be loaded by a word-processing program),
- reformatting your source questionnaire program into a typical-looking typed questionnaire document (it automatically numbers all the questions, draws boxes for the answers, specifies card and column locations for each answer in the data file, and writes skip and other instructions.), and
- generating a cross-reference listing of the skip instructions in your questionnaire program.

The CONVERT program also allows you to modify how these functions are performed. For example, when translating the QPL data file into a new format, such as Lotus, you can tell the CONVERT program which questions to include when it creates the spreadsheet. You can also tell it which records to select. These features allow you to use all, or just part, of your QPL data file in your spreadsheet. (See chapter 12.)

Language Introduction

QPL programs are written as a series of questions. Each question is equivalent to a single variable or a field in a data record. The language provides you with a variety of fundamental types of questions, such as multiple choice or short answer questions, which you can link together in complex arrangements through the use of skip instructions. The COLLECT, NCOLLECT, and KEYPUNCH programs will alter the data-entry method on the computer screen according to the type of question you specify in order to make it as easy as possible for the interviewer to enter the correct response.

Eight types of questions are displayed to the interviewer:

- multiple choice,
- check-all-that-apply,
- number,
- short answer,
- long, open-ended answer,
- 6-digit date,
- 8-digit date, and
- information only (no response recorded) questions.

In addition, four more types of questions are not displayed because the computer can answer them itself:

- time of interview,
- date of interview,
- version number of questionnaire, and
- a random number.

All 12 types of questions are programmed using the same basic syntax. The following example outlines the minimum information needed to program a single question.

Example

```
.QUESTION = variable name, TYPE = question type
    ...
    ...
    ...
.ANSWER
    ...
    ...
    ...
.NEXT
```

Note: Periods must be used to start command lines. The uppercase words are QPL command words, lowercase words specify the type of argument required, and the ellipses (...) indicate where the text of the question and answer are typed.

Computer Memory

A questionnaire program may contain from 1 to 1,056 questions. You may, however, run out of memory before this if your program contains many questions with large amounts of text, or when using certain memory-intensive functions, such as sorting records using the KEYPUNCH program or translating the data file from one version of your questionnaire to another using the CONVERT program. Although the QPL system programs will not use any expanded or extended memory your computer may have, you should be able to run programs with as many as 500 questions on computers with 640K bytes of lower memory without running into problems.¹

Exactly when a questionnaire program will run into memory problems depends upon many factors. The QPL system programs request memory from DOS only as they need it. The number of questions, skip instructions, compute statements, and answer and text lines all nibble away at the available memory. You should always test your program to make sure it will not have memory problems before giving it to a customer.²

Program Lines

Each program line can have one of five different purposes. It may be a command line, a question text line, an answer text line, a comment line, or a blank line. The QPL COMPILE program expects every line of your program to end with carriage return and line feed codes (called “[HRT]” in WordPerfect). This means that you should not depend on how your word processor automatically wraps lines according to your margin settings, but instead make sure that you have pressed Enter to end each line.

You should not use any of the special formatting features that your word processor may provide, such as underlining or bolding. The COMPILE program expects your program lines to contain only standard ASCII characters, and not any codes that are unique to your word processor.

¹The KEYPUNCH program uses expanded or extended memory only when sorting records.

²The ANSWER command may be used to conserve memory by reusing answer lists and format strings.

Command Lines

All command lines in a questionnaire must begin with a period in the first column of the line. QPL commands must be used to define a question's variable name and type, and where the question and answer text begin and end. More than one command may be typed on a line if the commands are separated by a comma or a blank space. Command words may be written in either uppercase or lowercase letters, and they can usually be abbreviated to just the first two characters of the word. (See appendix III.)

QUESTION and TYPE Commands

The QUESTION command defines the name of a question and where the text of a question begins. The TYPE command defines which of the 12 types of questions it is. These two commands must always be used at the beginning of each question.

Sample Questionnaire Program

```
*****
Comment line → ** PROGRAM: SAMPLE.DOC      DATE: 07/13/89 **
                ** AUTHOR: KEVIN DOOLEY          **
                **                               **
                ** TITLE: SAMPLE QPL QUESTIONNAIRE **
                *****
Command line → .CARD = 60, ESCAPE = FINISH
               .TITLE = "Sample QPL Questionnaire"

Question/next set → [.QUESTION = NAME, TYPE = STRING
                    What is your name?
                    .ANSWER = 15
                    .NEXT

                    .QUESTION = AGE, TYPE = NUMBER = "## years"
                    How old are you [NAME]?
                    .ANSWER
                    .NEXT

Blank line →

                    .QUESTION = GRAD, TYPE = DATE
                    When did you graduate from high school?
                    .ANSWER
                    660601
                    .NEXT

Question text line → .QUESTION = SEX, TYPE = MULT
                    What is your sex?
                    .ANSWER
Answer text line → Male
                  Female
                  .NEXT

                    .QUESTION = FINISH, TYPE = MULT
                    You have finished this questionnaire.
                    Do you want to review your answers?
                    .ANSWER
                    Yes
Skip instruction → .GOTO = NAME
                  No
                  .NEXT
```

ANSWER Command

The ANSWER command defines the end of the question text and the beginning of the answer text. Only multiple choice, check-all-that-apply, and random number questions may have more than one answer text line. The other types of questions use the number of characters in the answer text line to define the width of the variable field in the data file and its default answer.

NEXT Command

The NEXT command defines the end of the list of answers and the end of the question. It can also be used to specify which question to skip to if you do not want to go to the next question in the list. The short questionnaire on the previous page demonstrates how these commands are used.

Variable Names

Variable names may have from one to eight characters, but must begin with a letter (*i.e.*, A-Z). Numbers and some punctuation characters also may be used. No blank spaces, square brackets (*i.e.*, “[” or “]”), double quotes, commas, asterisks, or equal signs may be used in a variable name. Case is not significant. For example, the variable names “Q12” and “q12” are equivalent. Check the documentation of the analysis software you plan to use, such as SPSS or SAS, for any other limitations on naming variables.

Comment Lines

Comments may be included in your questionnaire program by typing an asterisk in the first column of a line. The entire line, up to the carriage return, is treated as a comment by the COMPILE program. Comments can also be added to the end of command lines by typing an asterisk, and then the comment.

Example

```
* This is a comment line.  
  
.NEXT = variable name * This is a comment on a command line.
```

Blank Lines

Blank lines (*i.e.*, lines with a carriage return in the first column) may also be used to make your program more readable. These lines may be used anywhere and are ignored by the compiler.

Question and Answer Text Lines

Text lines contain the actual text of a question or answer. They can begin with any character, except an asterisk, a carriage return, or a period. There are two limitations on how much text can be used in a question:

- Text lines cannot be more than 75 characters long and should be kept to between 30 and 40 characters for the most readability when displayed on the screen by the COLLECT program or when printed in questionnaire format by the CONVERT program.
- Up to 19 question and answer lines may be used in a single question. Only multiple choice, check-all-that-apply, and random questions allow more than one answer line; all other questions allow only one.

The KEYPUNCH program only displays the first line of a question at the top of the screen as the highlight bar is moved up and down the list of questions in the middle of the screen. The entire text of a question, however, will be displayed when the F2 key is pressed.

The question text lines will be used as variable labels by the CONVERT program when it generates SPSS or SAS statistical analysis programs. In this case, the question text lines will be merged together until the 40-character label limitation is reached.

References to Other Answers

Question text lines can be programmed to display the current answers from other questions. For example, if your first question asked for a respondent's name, a subsequent question can display his or her name in the text of the question.

Example

Q: What is your name?

A: John

Q: Hello John, ...

References to the current answers to questions are made by using the variable name of a question, delimited by brackets, in the text of a question. The COLLECT and KEYPUNCH programs will replace the variable name with the current answer to that question.

Types of Questions

QPL programs can ask seven types of questions that are answered by the interviewer: multiple choice, check-all-that-apply, short answer, open-ended answer, number, and long and short date questions. Four additional types of information can be automatically obtained by the computer with the COLLECT¹ and KEYPUNCH programs. These include the date and time that the interview took place, the version number of the questionnaire, and a random number that can be used to randomize the order in which questions are asked. They are programmed using the same syntax as questions. In addition to these, it has another type of question that displays information on the screen but does not record any data in the file. This is useful when you want to display information for the interviewer but do not need to record a response.

Complex questionnaires can be developed using these basic types of questions. This chapter describes how to program each type of question and how they work with the COLLECT, KEYPUNCH, and CONVERT programs.

Multiple Choice

Multiple choice questions are the backbone of a computer-aided telephone interview survey. Typically, they will make up the great majority of questions in a telephone interview.

Multiple choice questions provide more control over what the interviewer may enter for a response than any of the other QPL question types. For example, the interviewer

- may select only one of the choices and
- must select an answer before proceeding to the next question.

While these limitations put the burden on the questionnaire designer to identify all the reasonable answers to a question, including a “Don’t know” response, it pays off later by providing cleaner, more complete information on the question.

Multiple choice questions also have other features that help improve the accuracy of the data recorded by the interviewer. These questions can be used to make conditional skips to any question in the survey, according to which answer was selected. When the interviewer selects an answer, the COLLECT and KEYPUNCH programs will automatically skip to the question that has been linked to that answer. Thus, multiple choice questions can be used to screen respon-

¹The network version of the COLLECT program, called NCOLLECT, works identically to the COLLECT program except where specifically noted. See chapter 10, “Running the NCOLLECT and MONITOR Programs,” for more information on network-specific functions.

dents and direct them into appropriate lines of questions. Although the branching can become very complicated, the burden on the interviewer to skip to the correct question is eliminated because the COLLECT and KEYPUNCH programs do this automatically.²

Data quality is also improved with multiple choice questions because of how the COLLECT program works. The interviewer never has to take his or her eyes off the monitor when answering a multiple choice question. Answers are selected by using the up and down arrow keys to move a highlight bar to a particular choice. The interviewer always sees the answer he or she is about to choose, which should minimize the chance of a mistake. In addition, if the interviewer believes a mistake was made, he or she can always back up to that question to see what answer was selected and change it if necessary.

The KEYPUNCH program displays only one answer on the screen at a time. The interviewer changes the answer by toggling through the list. The answer that will be saved is always the one that is shown on the screen. As with the COLLECT program, the interviewer can always back up and change an answer if necessary.

Usage

```
.QUESTION = variable name, TYPE = MULT
.SUBTITLE = "phrase"
.PUT = variable name
.COPY = variable name
.CLOCK = ON or OFF
.FOREGROUND = color, BACKGROUND = color
.HIGHLIGHT = color, BORDER = color
.COMPUTE (expression)
.COMPUTEIF (test expression) (expression)
...
...
.ANSWER = default answer number, or = variable name
...
.GOTO = variable name
...
...
.IF (test expression) variable name
.SKIP (target expression)
.SKIPIF (test expression) (target expression)
.NEXT = variable name
```

Notes: Commands and arguments in bold type are required. The other commands are not required, but may be used to modify how the question works. Lowercase words indicate the type of argument a command requires and ellipses show where text lines should be typed.

Specifying Question Type

The TYPE command must be set to "MULT."

²See chapter 4, "Program Flow."

Answer Lines

At least one answer line is required. You can use up to 19 question and answer text lines. Each answer in the list, however, must fit on one line, and each line cannot be longer than 75 characters. If you need more responses than will fit on one screen, you can call one of the responses “More choices,” and use it to skip to a second question that has additional responses. Later, during analysis, you can merge the answers from the two questions into one variable.

Skip Instructions

Multiple choice questions can be used to direct respondents into particular lines of questions using the GOTO, IF, SKIP, SKIPIF, and NEXT commands. For example, if a respondent answered “Yes,” to a question, you could use a GOTO command to make the COLLECT and KEYPUNCH programs automatically jump to a particular question. A GOTO command is linked to a response by typing the command on a line immediately following the response.

Example

Source Program	Questionnaire Format
.QUESTION = Q1, TYPE = MULT	1. ...
...	...
...	...
...	...
.ANSWER	(CHECK ONLY ONE ANSWER)
Yes	
.GOTO = Q6	__ 1. Yes (GO TO QUESTION 6)
No	__ 2. No
Don't know	__ 3. Don't know
.NEXT = Q33	
	PLEASE SKIP TO QUESTION 33
	=====

In the above example, if the interviewer selects the answer “Yes,” the COLLECT program will skip to question Q6. If another answer is selected, the COLLECT program will jump to the question specified in the NEXT command. In this case, the COLLECT program will jump to question Q33. If the NEXT command had not specified a particular question, the COLLECT program would go to the next question in the questionnaire. The KEYPUNCH program will behave the same way. The column on the right shows how this question is printed, complete with skip instructions, by the CONVERT program.

Other skip commands may also be used in multiple choice questions. These are used to define more complex skip instructions. The responses to several questions can be evaluated in an IF command, for example, to determine which question should be displayed next. (See chapter 4.)

Default Answer

The ANSWER command can be used to specify which answer will be highlighted when the question is first displayed by the COLLECT and KEYPUNCH programs. If one answer will be commonly selected, you can save the interviewer some keystrokes by making that the default answer. Highlighting a particular question does not bias the response because the interviewer does not tell the respondent what answer is highlighted on the screen.

Example

```
.ANSWER = 3
```

In this example, the third answer in the list will be highlighted when displayed by the COLLECT and KEYPUNCH programs. If no answer is specified, the first answer in the list will be highlighted.

Copying Answer Lists

The ANSWER command can instead be used to copy the answers from a previous question, including the default highlight answer. Copying answers can save programming time and computer memory because the answer list needs to be written only once. In addition, if changes in the list need to be made, only the original list needs to be edited. The changes will be copied automatically.

Example

Source Program	Questionnaire Format
.QUESTION = Q1, TYPE = MULT	1. ...
...	...
...	...
...	...
.ANSWER	(CHECK ONLY ONE ANSWER)
Yes	
No	<input type="checkbox"/> 1. Yes
Don't know	<input type="checkbox"/> 2. No
.NEXT	<input type="checkbox"/> 3. Don't know
.QUESTION = Q2, TYPE = MULT	2. ...
...	...
...	...
...	...
.ANSWER = Q1	(CHECK ONLY ONE ANSWER)
.NEXT	
	<input type="checkbox"/> 1. Yes
	<input type="checkbox"/> 2. No
	<input type="checkbox"/> 3. Don't know

In this example, the answer list is automatically copied to question Q2 by setting its answer command equal to Q1. The right-hand column shows how this example would appear if it was printed in questionnaire format with the CONVERT program.

An answer list may not be copied if GOTO commands need to be linked to particular answers. You must retype the list and insert the GOTO commands. You may, however, copy an answer list that contains GOTO commands. In this case, only the list of answers is copied, not the GOTO commands.

Computing Responses

Instead of being used to input responses by the interviewer, multiple choice questions can be programmed to answer themselves, based on the responses to other questions. This ability has several useful applications. Since a multiple choice response can be displayed in the text of another question, you can use this feature to customize the text of a question for a particular respondent. You can summarize several other questions, categorize another question, or copy a response to another question and apply a new label.

The COMPUTE and COMPUTEIF commands can be used to automatically fill in the response as the questionnaire program is being used with the COLLECT or KEYPUNCH programs. The question, however, will not be displayed to the interviewer by the COLLECT program. Instead, when the interviewer reaches the question, the program will automatically evaluate the expressions, record the answer, and move to the next question (according to any skip instructions you may have used).

COMPUTE Command

The COMPUTE command must be used within the question it is being used to automatically answer, and it requires one argument that may be a single number or variable name, or complex expression. The COLLECT and KEYPUNCH programs will evaluate the argument and use the result for an answer. Since multiple choice questions code the answers as numbers, beginning at 1, your expression should evaluate to the number of one of the answers in the list.

Example

Source Program	Questionnaire Format
.QUESTION = Q1, TYPE = MULT	1. ...
...	
.ANSWER	(CHECK ONLY ONE ANSWER)
Male	
Female	__ 1. Male
.NEXT	__ 2. Female
.QUESTION = Q2, TYPE = MULT	2. ...
.COMPUTE (Q1)	
...	(CHECK ONLY ONE ANSWER)
.ANSWER	
Mr.	__ 1. Mr.
Ms.	__ 2. Ms.
.NEXT	
	COMPUTE #1

Chapter 3 Types of Questions

```
.QUESTION = Q3, TYPE = VOID      3. Hello #2 ...
Hello [Q2] ...
.ANSWER                          Press ENTER to continue...
Press ENTER to continue...
.NEXT
```

In this example, a COMPUTE command is used to automatically answer question Q2. After the interviewer answers question Q1, the next question he or she will see on the screen will be question Q3. The COMPUTE command will automatically copy the number of the answer that will be coded for question Q1. Since question Q1 only has two answers, it can only have a value of 1 or 2, which matches the values that question Q2 may have. If the interviewer selects the answer “Female,” the answer to question Q2 will be coded as a 2. Then, when the answer to Q2 is used in the text of question Q3, it will be displayed as “Hello Ms...”

Make sure that your expressions evaluate to a number that will be in your list of answers. If it evaluates to a higher number, a number will be displayed instead of an answer from the list when using it in the text of another question. The following example shows how a COMPUTE statement could be used with a more complex expression.

Example

Source Program

```
.QUESTION = BDATE, TYPE = DATE
When were you born?
.ANSWER
.NEXT

.QUESTION = BDAY, TYPE = MULT
.COMPUTE (JDAYOFWEEK( TOJUL( BDATE )))
Calculate day of week of birth.
.ANSWER
Sunday
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
.NEXT

.QUESTION = Q32, TYPE = VOID
You were born on a [BDAY].
.ANSWER
Press ENTER to continue...
.NEXT
```

Chapter 3 Types of Questions

Questionnaire Format

1. When were you born?

19|__|__|-|__|__|-|__|__|
Year Month Day

2. Calculate day of week of birth.

(CHECK ONLY ONE ANSWER)

|__| 1. Sunday
|__| 2. Monday
|__| 3. Tuesday
|__| 4. Wednesday
|__| 5. Thursday
|__| 6. Friday
|__| 7. Saturday

COMPUTE (JDAYOFWEEK(TOJUL(#1)))

3. You were born on a #2.

Press ENTER to continue...

In this example, two date functions are used with a COMPUTE command to find the day of the week on which the respondent was born, and record it as an answer to question BDAY.³ The TOJUL function returns the Julian date of the date that was entered as a response to the date question INDATE. Then the JDAYOFWEEK function converts the Julian date to a number that represents the day of the week. The answer list is ordered to match the values returned by the JDAYOFWEEK function (Sunday is 1, and Monday is 2, etc.).

After the interviewer enters the birthday, the COLLECT program will calculate the day of the week, and then it will display question Q32. If the interviewer entered the date January 1, 1971 to BDATE, for example, question Q32 will be displayed as “You were born on a Saturday.”

If the interviewer backs up and changes the answer to BDATE, the answer to BDAY will be recomputed when the interviewer goes forward through the question again. You should only use questions in COMPUTE and COMPUTEIF expressions that have been answered by the interviewer. Questions that are off the current skip path, or that have not yet been reached, all contain default values. There is no guarantee that these questions will always have the default values you specified in your program because an interviewer may go down a path, entering new answers, but then back up and go down a different path. The

³See chapter 4, “Functions.”

answers that were entered to questions that are now off the skip path are not dropped until the interview record is added to the data file.⁴

COMPUTEIF Command

The COMPUTEIF command works just like the COMPUTE command, but it also allows you to use a test expression to determine when a computed value should be used. Any number of COMPUTE and COMPUTEIF commands may be used within a single question, but the COLLECT and KEYPUNCH programs will stop evaluating these commands after the test expression for a COMPUTEIF command evaluates to a true value. The COMPUTEIF command is commonly used, for example, to categorize one or more questions.

Example

Source Program	Questionnaire Format
<pre>.QUESTION = Q1 .TYPE = NUMBER = "## years" How old are you? .ANSWER .NEXT</pre>	<pre>1. How old are you? _ _ years</pre>
<pre>.QUESTION = Q2, TYPE = MULT .COMPUTEIF (Q1 < 18) (1) .COMPUTEIF (Q1 < 30) (2) .COMPUTE (3) Put age into one of three categories. .ANSWER young middle aged old .NEXT</pre>	<pre>2. Put age into one of three categories. (CHECK ONLY ONE ANSWER) _ 1. young _ 2. middle aged _ 3. old COMPUTE IF (#1 < 18) 1 COMPUTE IF (#1 < 30) 2 COMPUTE 3</pre>
<pre>.QUESTION = Q3, TYPE = VOID We have determined that you are [Q2]. .ANSWER Press ENTER to continue... .NEXT</pre>	<pre>3. We have determined that you are #2. Press ENTER to continue...</pre>

In this example, a combination of COMPUTE and COMPUTEIF commands were used in question Q2 to categorize the age of the respondent. The COMPUTE and COMPUTEIF commands will be evaluated in the order that they were used in a question. For example, if a respondent said that he or she was 15 years old, the test expression for the first COMPUTEIF command will be true, so the answer to question Q2 will be set to 1, and the text of question Q3 will say “We have determined that you are young.”

⁴See chapter 5, “Missing Values.”

If the respondent said that he or she was 25 years old, the first COMPUTE command will be false, which will cause the COLLECT program to process the next COMPUTE command. Since the second test expression will be true, the answer to question Q2 will be set to 2, and the text of the last question will say “We have determined that you are middle aged.”

Finally, if the respondent is 30 years old or older, the test expressions for both COMPUTE commands will be false. In this case, the COLLECT program will use the last COMPUTE command to set the answer to 3, and the text of the last question will say “We have determined that you are old.”

Check-All-That-Apply

Check-all-that-apply questions allow the interviewer to select more than one response from a list of answers, or to select none of the answers. While they are programmed similar to multiple choice questions, they are different because multiple choice questions only allow one answer to be selected and they require that an answer be selected before the interviewer may move to the next question.

When this question is displayed on the screen by the COLLECT program, the interviewer can move a highlight bar up and down the list of answers. The interviewer can select an answer by first highlighting it, and then pressing the space bar. A check mark will be displayed next to the answer to indicate that it has been selected. The check mark may be toggled on and off by highlighting it and pressing the space bar again. The KEYPUNCH program also displays the list of answers and shows which have been checked when the interviewer highlights the question on the edit screen. Both programs will display a message to the interviewer at the bottom of the screen that tells them to use the space bar to check responses.⁵

Usage

```
.QUESTION = variable name, TYPE = CHECK
.SUBTITLE = "phrase"
.PUT = variable name
.COPY = variable name
.CLOCK = ON or OFF
.FOREGROUND = color, BACKGROUND = color
.HIGHLIGHT = color, BORDER = color
.COMPUTE (expression)
.COMPUTEIF (test expression) (expression)
...
...
...
```

⁵You may change the text of this message by using QPL's foreign language error message capabilities. Instead of translating this message to another language, you can replace it with your own message. See chapter 7, “Translating QPL Error Messages.”

Chapter 3 Types of Questions

```
.ANSWER = variable name
...
.GOTO = variable name
...
...
.IF (test expression) variable name
.SKIP (target expression)
.SKIPIF (test expression) (target expression)
.NEXT = variable name
```

Notes: Commands and arguments in bold type are required. The other commands are not required, but may be used to modify how the question works. Lowercase words indicate the type of argument a command requires and ellipses show where text lines should be typed.

Specifying Question Type

The TYPE command must be set to “CHECK.”

Answer Lines

At least one answer line is required. You can use up to 19 question and answer text lines. Each answer in the list, however, must fit on one line. If you need more answers than will fit on one screen, you can call one of the responses “More choices,” and use it to skip to a second question that has additional responses.

When the response to this type of question is used in the text of another question, only the numbers of the answers that were checked are listed, instead of the full text of each question.

Example

Source Program	Questionnaire Format
<pre>.QUESTION = Q1, TYPE = CHECK What are your favorite colors? .ANSWER Red Orange Yellow Green Blue Indigo Violet .NEXT</pre>	<pre>1. What are your favorite colors? (CHECK ALL THAT APPLY) __ 1. Red __ 2. Orange __ 3. Yellow __ 4. Green __ 5. Blue __ 6. Indigo __ 7. Violet</pre>
<pre>.QUESTION = Q2, TYPE = VOID Your favorite colors are [Q1]. .ANSWER Press ENTER to continue... .NEXT</pre>	<pre>2. Your favorite colors are #1. Press ENTER to continue...</pre>

If the interviewer checks Red, Orange, and Yellow, the text of question Q2 will be displayed as “Your favorite colors are 1, 2, 3.”

Copying Answer Lists

If you will be using the same list of answers for two or more check-all-that-apply questions, you can use the ANSWER command to tell the COMPILE program to copy the answer list from a previous question. Copying the answer lists can save programming time and computer memory because the answer list needs to be written only once. If changes need in the list need to be made while you are developing your questionnaire, only the original list needs to be edited. The changes will be copied to the other questions automatically.

Example

Source Program	Questionnaire Format
.QUESTION = Q1, TYPE = CHECKANSWER Red Orange Yellow .NEXT	1. ... (CHECK ALL THAT APPLY) <input type="checkbox"/> 1. Red <input type="checkbox"/> 2. Orange <input type="checkbox"/> 3. Yellow
.QUESTION = Q2, TYPE = CHECKANSWER = Q1 .NEXT	2. ... (CHECK ALL THAT APPLY) <input type="checkbox"/> 1. Red <input type="checkbox"/> 2. Orange <input type="checkbox"/> 3. Yellow

Skip Instructions

Check-all-that-apply questions can be used to direct respondents into particular lines of questions using the GOTO, IF, SKIP, SKIPIF, and NEXT commands. For example, if a respondent said that they had an "Other" response, you could use the GOTO command to skip to a question where it could be described.

Example

Source Program	Questionnaire Format
.QUESTION = Q1, TYPE = CHECK What are your favorite colors? .ANSWER Red Orange Yellow Green Blue Indigo Violet Other .GOTO = Q2 .NEXT = Q3	1. What are your favorite colors? (CHECK ALL THAT APPLY) <input type="checkbox"/> 1. Red <input type="checkbox"/> 2. Orange <input type="checkbox"/> 3. Yellow <input type="checkbox"/> 4. Green <input type="checkbox"/> 5. Blue <input type="checkbox"/> 6. Indigo <input type="checkbox"/> 7. Violet <input type="checkbox"/> 8. Other (GO TO QUESTION 2) SKIP TO QUESTION 3 =====

Chapter 3
Types of Questions

```
.QUESTION = Q2, TYPE = STRING      2. What is your other favorite
What is your other favorite         color?
color?
.ANSWER = 10                        |_|_|_|_|_|_|_|_|_|_|_|_|_|_|
.NEXT
```

In this example, a GOTO command has been linked to the response called “Other.” If it is checked by the interviewer, the COLLECT program will skip to question Q2, where the interviewer may type in the name of the respondent’s favorite color. If this response is not checked, the COLLECT program will skip to the question specified by the NEXT command, Q3, since no other GOTO commands have been used.

Using more than one GOTO command with a check-all-that-apply question is difficult since, unlike multiple choice questions, more than one answer may be checked. By definition, the COLLECT and KEYPUNCH programs always will use the first GOTO command that is linked to an answer that is checked.

The IF command should be used when you need to branch to more than two different questions, or when you need to base a skip on a particular combination of answers. Using this command, you can write a series of expressions that can test for any combination of responses to determine which question should be asked next.

Example

Source Program	Questionnaire Format
.QUESTION = Q1, TYPE = CHECK	1. What are your favorite colors?
What are your favorite colors?	(CHECK ALL THAT APPLY)
.ANSWER	
Red	_ 1. Red
Orange	_ 2. Orange
Yellow	_ 3. Yellow
Green	_ 4. Green
Blue	_ 5. Blue
Indigo	_ 6. Indigo
Violet	_ 7. Violet
.IF (Q1 @ 2 OR Q1 @ 3) Q2	
.IF (Q1 @ 4) Q3	
.IF (Q1 = 0) Q4	
.NEXT = Q5	IF (#1 @ 2 OR #1 @ 3) GO TO #2
	IF (#1 @ 4) GO TO #3
	IF (#1 = 0) GO TO #4
	SKIP TO QUESTION 5
	=====

In this example, IF commands are used to program more complex skips. IF commands are always evaluated in the order that they are listed, and the COLLECT and KEYPUNCH programs will stop processing them after one evaluates to a true value. Since the answers to check-all-that-apply questions are stored in a binary format (each answer is coded 0 if it is not checked, or 1 if it is checked) you must use the @ operator to determine whether particular answers have been checked.⁶

The first expression will evaluate to a true value if the second answer (Orange) or the third answer (Yellow) were checked. If either of these answers was checked, the COLLECT program will skip to question Q2. If neither of the answers was checked, the COLLECT program will evaluate the next IF statement. It will skip to question Q3 if the fourth answer (Green) is checked. If the fourth answer is not checked (and the second and third answers are not checked), the COLLECT program will evaluate the next IF command.

The third IF expression will be true if none of the answers were checked. In this case, all of the individual answers will be coded 0, making the binary value of the field zero. If one of the answers was checked, however, the binary value of the field will not be zero, so the COLLECT program will skip to the question specified by the NEXT command, Q5.

Computing Responses

Computing responses for check-all-that-apply questions, while possible, is rarely very useful. Since the response to this question is displayed as a list of numbers of the answers that were selected (when used in the text of another question), there seems to be little reason to use this capability. If you need to create one or more phrases that are customized to particular respondents, you should instead use multiple choice questions and the COMPUTE and COMPUTEIF commands.

Since the responses to check-all-that-apply questions are stored in a binary format, computing the correct value that will check particular answers requires some care. You must use the formula, " $2^{m-1} + 2^{n-1} + \dots$," (where m and n are the numbers of the answers you wish to check) to check answers. The following example shows how a COMPUTE command could be written to check the second and third answers of a question.

⁶See chapter 4, "Conditional Skips," for more information about IF commands and the QPL equation processor. See chapter 5 for information about how such questions are stored in the data file.

Example

Source Program	Questionnaire Format
<pre>.QUESTION = Q1, TYPE = CHECK What are your favorite colors? .ANSWER Red Orange Yellow Green Blue Indigo Violet .COMPUTE (2^(2-1) + 2^(3-1)) .NEXT</pre>	<pre>1. What are your favorite colors? (CHECK ALL THAT APPLY) __ 1. Red __ 2. Orange __ 3. Yellow __ 4. Green __ 5. Blue __ 6. Indigo __ 7. Violet COMPUTE (2^(2-1) + 2^(3-1))</pre>

In this example, the QPL equation processor will be used to calculate the decimal equivalent of the binary number that will check the second and third answers: Orange and Yellow. Note that, in this case, the expression could have been simplified to 6, which is the decimal equivalent to the binary number 110.

Number

The number question type has several features that help the interviewer enter the correct number when using the COLLECT or KEYPUNCH programs.

- Only numbers may be entered. The COLLECT and KEYPUNCH programs will display an error message and beep if a non-number key is pressed.
- Any special number format can be used, such as telephone numbers, social security numbers, or general numbers with commas and a decimal point.
- The range of values a correct number may have can be specified in your QPL questionnaire. If a number is out-of-range, an error message will be displayed at the bottom of the screen and the computer will beep. The interviewer cannot move to another question until a number has been entered that lies within your given range.
- “Don’t know” responses may be entered by pressing the question mark key. A question mark will be displayed on the screen instead of a number.

Usage

```
.QUESTION = variable name
.TYPE = NUMBER = decimal places or = "format string"
.SUBTITLE = "phrase"
.PUT = variable name
.COPY = variable name
.CLOCK = ON or OFF
.FOREGROUND = color, BACKGROUND = color
.HIGHLIGHT = color, BORDER = color
.HIGH = (number or expression)
.LOW = (number or expression)
.COMPUTE (expression)
.COMPUTEIF (test expression) (expression)
...
...
...
.ANSWER = variable name
...
.IF (test expression) variable name
.SKIP (target expression)
.SKIPIF (test expression) (target expression)
.NEXT = variable name
```

Notes: Commands and arguments in bold type are required. The other commands are not required, but may be used to modify how the question works. Lowercase words indicate the type of argument a command requires, and ellipses show where text lines should be typed.

Specifying Question Type

The TYPE command must be set to "NUMBER."

The NUMBER argument can take a sub-argument that specifies the number of decimal places or a special format for the number. You may specify a number that has up to nine decimal places. If you want to use an integer, you may either set the number of decimal places to zero or omit specifying any decimal places.

Example

Source Program	Questionnaire Format
.QUESTION = variable name	1. ...
.TYPE = NUMBER = 2	...
...	...
...	
...	_ _ . _ _
.ANSWER	
0000	
.NEXT	

Formatted Numbers

You may also design any type of format for a number by specifying a format string. This allows you to show numbers with dollar signs, commas, a decimal point, dashes, or any other characters. Formatting a number makes it easier for the interviewer to read the number that he or she is keying.

Format strings must be delimited by double quotes. The character “#” indicates where numbers may be typed. A period marks the decimal place, and commas are used as thousands separators.⁷ Other characters, including blank spaces, may be used to organize a particular type of number.

Example

Source Program	Questionnaire Format
.QUESTION = Q1	1. ...
.TYPE = NUMBER = "\$#,###.##"	...
...	...
...	
...	\$ _ , _ _ _ · _ _
.ANSWER	
.NEXT	

In this example, question Q1 is a six-digit number, including two decimal places. The COMPILE program automatically calculates how many decimal places a number has from the format string. This format string also specifies that the number be displayed by the COLLECT and KEYPUNCH programs with a dollar sign and a comma. The column on the right shows how this number will be printed in questionnaire format by the CONVERT program. The COLLECT and KEYPUNCH programs record only the numbers that the interviewer types in. The dollar sign, comma, and decimal point are not saved in the data file. They are displayed only for the convenience of the interviewer.

Field Size

You must specify how many digits your largest possible number may have. Number questions may have from 1 to 15 digits, including up to 9 decimal positions and 1 position for a minus sign. The largest range of numbers that may be entered extends from -99,999,999,999,999 to 999,999,999,999,999.⁸ Generally, you should use the fewest digits possible in order to keep the data file compact, save computer memory, and limit the range of numbers that may be entered.

There are four ways to define how many digits your largest number may have. You may define the field size using

- a default answer text line,
- a format string,
- the ANSWER command, or
- a copy of the format of a previous number question.

⁷It is customary in some countries to use other characters as a decimal point and thousands separators. QPL programs use the country setting in your CONFIG.SYS file to determine which characters should be used. See chapter 7.

⁸Numbers written in scientific notation are not allowed.

Chapter 3
Types of Questions

Example

Source Program	Questionnaire Format
.QUESTION = Q1, TYPE = NUMBERANSWER 12345 .NEXT	1. _ _ _ _ _
.QUESTION = Q2 .TYPE = NUMBER = "###.##"ANSWER .NEXT	2. _ _ _ _ . _ _ _
.QUESTION = Q3 .TYPE = NUMBER = 2ANSWER = 5 .NEXT	3. _ _ _ _ . _ _ _
.QUESTION = Q4 .TYPE = NUMBERANSWER = Q2 .NEXT	4. _ _ _ _ . _ _ _

In the above example, question Q1 is set to a five-digit number with no decimal places. A default answer was used here to set the number of digits. The default answer, 12345, will be shown in the answer field when the question is displayed on the screen by the COLLECT and KEYPUNCH programs, where the interviewer may change it. (Note: If the number was defined at the TYPE command to have two decimal places, the default answer would be displayed as "123.45." The number of decimal places is always set at the TYPE command, and not at the default answer line. The decimal point is not written in the default answer line.)

Question Q2 is a five-digit number with two decimal places. The number of digits was defined in the format string, making it unnecessary to include a default answer. The default answer for this question will be zero when it displayed by the COLLECT and KEYPUNCH programs.

Question Q3 sets the number of digits using the ANSWER command. This method requires less memory than using a format string because no format string needs to be stored.

Finally, question Q4 shows how the format information may be copied from a previous question using the ANSWER command. Here, both the number of digits and decimal places will be copied from question Q2.

Range of Values

The highest and lowest number that may be entered is automatically limited by the number of digits in the field. For example, if you define a four-digit question, which includes two decimal places, the lowest number the COLLECT and KEYPUNCH programs will allow to be entered is -9.99, and the highest number is 99.99.

You may further limit the range of values by using the HIGH and LOW commands. You may use either, neither, or both of these commands to specify the highest and lowest values a particular question may have. These values may be either a constant number, the current value of another question, or an expression that includes both constants and other questions. If the interviewer tries to enter a number that is not within this range, the COLLECT and KEYPUNCH programs will display an error message and beep. The interviewer cannot move to another question until he or she enters an acceptable value.

Example

Source Program	Questionnaire Format
.QUESTION = Q1	1. ...
.TYPE = NUMBER = 2	...
.LOW = 0, HIGH = 20.0	...
...	_ _ . _ _
...	
.ANSWER = 4	LOWEST VALUE = 0
.NEXT	HIGHEST VALUE = 20
.QUESTION = Q2	2. ...
.TYPE = NUMBER = 2	...
.HIGH = Q1	...
...	_ _ . _ _
...	
.ANSWER = 4	HIGHEST VALUE = #1
.NEXT	

Chapter 3 Types of Questions

```
.QUESTION = Q3          3. ...  
.TYPE = NUMBER = 2     ...  
.LOW = (Q1 + 1)        ...  
...  
...                     |__|__|.|__|__|  
...  
.ANSWER = 4            LOWEST VALUE = (#1 + 1)  
.NEXT
```

In the above example, question Q1 is a four-digit number, including two decimal places. The LOW value is set to 0 and the HIGH value is set to 20.00. If the interviewer tries to enter a number such as -0.98 or 33.43, the COLLECT and KEYPUNCH programs will display an error message. The interviewer, however, may still enter a “Don’t know” response (a question mark) and move to the next question despite these limits.

Question Q2 only has a limit on the highest value that the interviewer may enter. The HIGH value has been set to the number that will be entered for question Q1 by the interviewer. Thus, the highest value that may be entered for Q2 is 20 (the highest value Q1 can have). If, however, the interviewer entered the number 15.00 for question Q1, the highest value that could be entered in question Q2 would also be 15.00. Since the LOW command was not used, the interviewer may enter a negative value up to -9.99.

An expression has been used to limit the lowest value that Q3 may have. Here, the lowest value is equal to the value of Q1 plus one. Note that parentheses must be used to delimit expressions.⁹

There are several important points that you should keep in mind when basing ranges on the values of other questions.

- You should refer only to questions that have already been answered.
- Both questions must be on the same path (*i.e.*, the interviewer should not be able to skip around the first question).
- You should not specify a high and low value from the same question.
- If the interviewer entered a “Don’t know” value for the question, the range check is not made.

⁹See chapter 4 for a more information on expressions.

Copying a Field Definition

The ANSWER command can be used to copy the field size, number of decimal places, default answer, and format string from a previous question. Copying answers can save programming time and computer memory because this information needs to be written only once. In addition, if changes need to be made, only the original answer and format string need to be edited. The changes will be copied automatically by the COMPILE program.

Example

Source Program	Questionnaire Format
.QUESTION = Q1	1. ...
.TYPE = NUMBER = "(###)###-####"	...
...	...
...	(_ _ _) _ _ _ _ - _ _ _ _ _
.ANSWER	
.NEXT	
.QUESTION = Q2, TYPE = NUMBER	2. ...
...	...
...	...
...	...
.ANSWER = Q1	(_ _ _ _) _ _ _ _ - _ _ _ _ _
.NEXT	

In this example, question Q1 is a telephone number. Question Q2 is also a telephone number because its ANSWER command was set to Q1. Since no default answer was specified in Q1, both questions will have default answers of zero when they are displayed by the COLLECT and KEYPUNCH programs. The column on the right shows how these questions are printed in questionnaire format by the CONVERT program.

Computing Responses

Instead of being used to input numbers from the interviewer, number questions can be programmed to answer themselves, using the COMPUTE and COMPUTEIF commands. You could, for example, use these commands to compute the sum of the responses to several other number questions or to calculate a percentage. You can display the results, using any number format, in the text of a subsequent question or use the results as the basis of a skip instruction.

Example

Source Program	Questionnaire Format
.QUESTION = Q1	1. ...
.TYPE = NUMBER = "##,###"	...
...	
...	_ _ _ , _ _ _ _
.ANSWER	
.NEXT	

Chapter 3
Types of Questions

```

.QUESTION = Q2, TYPE = NUMBER      2. ...
...
...
.ANSWER = Q1                        |_|_|_|, |_|_|_|_|
.NEXT

.QUESTION = Q3, TYPE = NUMBER      3. ...
...
...
.ANSWER = Q1                        |_|_|_|, |_|_|_|_|
.NEXT

.QUESTION = Q4                      4. ...
.TYPE = NUMBER = "#,###,###"      ...
...
...
.ANSWER                             |_|_|_|_|_|, |_|_|_|_|_|
.COMPUTE (Q1 + Q2 + Q3)           COMPUTE (#1 + #2 + #3)
.NEXT

.QUESTION = Q5                      5. ...
.TYPE = NUMBER = "##.##%"        ...
...
...
.ANSWER                             |_|_|_|. |_|_|%
.COMPUTE (Q1 / Q4 * 100)         COMPUTE (#1 / #4 * 100)
.NEXT

.QUESTION = Q6, TYPE = VOID        6. The sum is #4.
The sum is [Q4].

The first answer, [Q1], is [Q5]    The first answer, #1, is #5
of the total.                      of the total.
.ANSWER                             Press ENTER to continue...
Press ENTER to continue...
.NEXT

```

In this example, a COMPUTE command is used in question Q4 to sum the previous three questions. The interviewer will not see this question on the screen. The COLLECT program will automatically compute the answer and move to the next question, according to any skip instructions you may have added. Since no skip instructions were used, it will go to the next question in the list.

A COMPUTE command has been used in question Q5 to calculate a percentage. This question also will not be displayed to the interviewer. After the response is automatically calculated, the COLLECT program will go to the next question.

Question Q6 will be the next question the interviewer sees after he or she answers question Q3. Question Q6 has been programmed to display several of the previous responses. If the sum, for example, was 200,000, and the response to question Q1 was 10,000, this question would be displayed as shown in the following example.

Example

```
The sum is 200,000.

The first answer, 10,000, is 5.0%
of the total.

Press ENTER to continue...
```

Similarly, you could use the COMPUTEIF command to perform more complex calculations. You could use it, for example, to test the value of one or more questions before computing a result.

Example

Source Program	Questionnaire Format
.QUESTION = Q7, TYPE = NUMBER	7. ...
...	...
...	...
.ANSWER = Q4	_ , _ _ _ , _ _ _
.COMPUTEIF (Q4 < 10000) (Q1)	
.COMPUTEIF (Q4 < 100000) (Q1 + Q2)	COMPUTE IF (#4 < 10000) #1
.COMPUTE (Q4)	COMPUTE IF (#4 < 100000) (#1 + #2)
.NEXT	COMPUTE #4

In this example, a combination of COMPUTE and COMPUTEIF commands are used to automatically calculate a response to question Q7. The compute commands will be evaluated in order and the COLLECT and KEYPUNCH programs will stop processing the commands when the test expression for a COMPUTEIF command is true. The test expression for the first COMPUTE will be true if question Q4 (the sum of the first three questions) is less than 10,000. If it is, the response to question Q1 will be used as a response. If it is not true, the next COMPUTEIF command will be evaluated. If its test expression is true, then the sum of the first two answers will be used as a response. If it is not true, the last COMPUTE command will be evaluated. Here, it just uses the answer to question Q4.

Rounding and Field Overflow

Although the QPL equation processor maintains 15-digit precision when evaluating an expression, it will round the final result, if necessary, in order to make it fit into the field you have set for a question. If a number is too small, it may be rounded to zero. If a number is too large to fit in a field, the field will be set to the “Don’t know,” or missing, response. You should make sure that your field contains enough decimal places for small numbers, or enough digits to accommodate the largest number that may be computed (including one position for a minus sign if you are using negative numbers).

Short Answer

Short answer (string) questions may be used to enter such items as names, short phrases, or other information that contains letters and numbers. The COLLECT and KEYPUNCH programs have several features that help the interviewer enter the correct response:

- Letters that the interviewer types can be automatically capitalized, regardless of whether the shift key is pressed or not.
- The highest and lowest allowable answers (compared alphabetically) can be specified in your QPL questionnaire. If the interviewer enters a response that is out of this range, the COLLECT and KEYPUNCH programs will display an error message and beep. The interviewer cannot move to another question until an allowable answer has been entered.
- You may use a formatted string to control where the interviewer may type letters and numbers in the field, and to make it clear what information should be entered in the field.
- The interviewer may enter a “Don’t know” response by typing a question mark in the first column of the answer line. The COLLECT and KEYPUNCH programs will ignore any range specifications and move to the next question if a “Don’t know” response is entered.

Usage

```
.QUESTION = variable name, TYPE = STRING = "format" or WORKSTATION
.SUBTITLE = "phrase"
.PUT = variable name
.COPY = variable name
.CLOCK = ON or OFF
.FOREGROUND = color, BACKGROUND = color
.HIGHLIGHT = color, BORDER = color
.LOW = "lowest allowable value"
.HIGH = "highest allowable value"
.UPPERCASE = ON or OFF
...
...
...
.ANSWER = field size or = variable name
...
.IF (test expression) variable name
.SKIP (target expression)
.SKIPIF (test expression) (target expression)
.NEXT = variable name
```

Notes: Commands and arguments in bold type are required. The other commands are not required, but may be used to modify how the question works. Lowercase words indicate the type of argument a command requires, and ellipses show where text lines should be typed.

Specifying Question Type

The TYPE command must be set to "STRING."

Formatted Answers

The TYPE command can also take a sub-argument that defines a format, or template, that limits where the interviewer may type letters and numbers in the field. If a format is not specified, the COLLECT and KEYPUNCH programs will allow the interviewer to type anything, anywhere in the field. String formats are programmed similarly to number formats:

- The format string must be delimited by double quotes.
- "#" defines where a number may be typed (0-9).
- "@" defines where a letter may be typed (A-Z).
- "*" defines where any character may be typed.

Any other characters, including blank spaces, may be used to indicate to the interviewer what information should be entered.

Example

```
Source Program

.QUESTION = Q1
.TYPE = STRING = "State: @@ Zip: #####"
...
...
...
.ANSWER
.NEXT

Questionnaire Format

1. ...
...
...

State: |_|_| Zip: |_|_|_|_|_|
```

In this example, question Q1 is a seven-character string question. The format string defines the size of the field. The COLLECT and KEYPUNCH programs will allow only letters to be typed in the first two characters of the field, and only numbers to be typed in the last five characters of the field, corresponding to the state abbreviation and zip code. Only the characters entered by the interviewer will be saved in the data file.

Workstation Name

A second sub-argument, WORKSTATION, can be used to tell the NCOLLECT program to use an interviewer's network account name as answer to a string question. This lets you automatically identify who is using the questionnaire. If the NCOLLECT program is running on a Novell, or compatible, local area network, it will

not display this question to the interviewer but instead copy the account name and then immediately go to the next question (as defined by any skip instructions).

The WORKSTATION sub-argument also sets the default field size to 45 characters, the maximum size of Novell account name. Thus, you do not need to specify a field size at the ANSWER command. You may, however, use the ANSWER command to shorten the field size if names on your system are not this long. Names longer than your setting will be truncated.

Example

Source Program

```
.QUESTION = Q1, TYPE = STRING = WORKSTATION
Copy interviewer's account name.
.ANSWER = 10
.NEXT

.QUESTION = Q2, TYPE = VOID
Your network account name is [Q1].
.ANSWER
Press ENTER to continue...
.NEXT
```

Questionnaire Format

1. Copy interviewer's account name.
|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|
2. Your network account name is #1.
Press ENTER to continue...

In this example, a string question is used to hold the interviewer's network account name. The ANSWER command was used to shorten the field to 10 characters. Instead of displaying question Q1 to the interviewer, the NCOLLECT program will store the name and then display question Q2 since no specific skip instructions were used. Q2 is then used to display the account name as part of the question text.

The WORKSTATION sub-argument may also be used when you run your questionnaire with the COLLECT or KEYPUNCH programs. Instead of using the network account name, these programs will copy the workstation name that was set as a command line option (/W) when they were started.

Example

```
C:\> COLLECT JOB1.QPL /WSmith
```

In this example, the COLLECT program is started from the DOS command line using a questionnaire program called JOB1.QPL. The /W option tells the COLLECT program to use the name “Smith” as this interviewer’s current account name. Thus, “Smith” would be displayed in question Q2 in the previous example.

The /W command line option also may be used with the NCOLLECT program to override the interviewer’s network account name with another name.

Field Size

String questions may have a field size ranging from 1 to 32,000 characters. Generally, you should keep the number of characters to a minimum in order to conserve computer memory and disk space. Also, few statistical programs, including Lotus and dBase, can use handle strings that are much longer than 200 characters.¹⁰ If you set the field size to a value greater than 80 characters, you must increase the default data card size that is used for the data files created by the COLLECT and KEYPUNCH programs.¹¹

There are four ways to set the field size for string questions:

- use a default answer line,
- set the number of characters at the ANSWER command,
- use a format string, or
- copy the format of a previous string question.

Example

Source Program	Questionnaire Format
.QUESTION = Q1, TYPE = STRING	1. ...
...	...
...	...
...	...
.ANSWER	_ _ _
abc	
.NEXT	
.QUESTION = Q2, TYPE = STRING	2. ...
...	...

¹⁰SAS string fields are limited to 200 characters (only the first 16 characters are significant in analytical procedures), SPSS/PC+ is limited to 255 characters (only the first 8 characters are significant in analytical procedures), Lotus is limited to 240 characters, and dBase and BASIC are limited to 254 characters.

¹¹See chapter 5, CARD command.

```
...
...
.ANSWER = 5
.NEXT

.QUESTION = Q3
.TYPE = STRING = "State: @@"
...
...
...
.ANSWER
.NEXT

.QUESTION = Q4, TYPE = STRING
...
...
...
.ANSWER = Q3
.NEXT
```

...
|_|_|_|_|_|

3. ...
...
...
State: |_|_|

4. ...
...
...
State: |_|_|

In this example, a default answer line is used in question Q1 to create a three-character string field. When this question is displayed on the screen by the COLLECT and KEYPUNCH programs, the default answer “abc” will be displayed in the answer field, where it may be changed by the interviewer. Only one default answer line may be in a string question. It may contain any combination of blank spaces and characters, but a carriage return must be typed immediately after the last character in the phrase. It may be useful to use a default phrase if the interviewer will not have to change the phrase very often. For example, if a question asks for an interviewer’s initials, setting the default phrase to his initials saves him the trouble of typing them in each time and increases the likelihood that they will be entered the same way in every interview. If another interviewer uses the program, he or she can erase the default initials and insert his own.

Question Q2 has been set to a five-character string field using the ANSWER command. Since a default answer line was not used, the field will be filled with blank spaces when it is displayed on the screen by the COLLECT and KEYPUNCH programs. You must use the ANSWER command to set the field size if you need to make the field larger than 75 characters long, because of the COMPILE program’s 75-character limitation on the size of question and answer text lines.

A format string was used to set question Q3 to a two-character string field. (Note: A two-character default answer line could also be used here to initialize the field to something other than blank spaces.) Question Q4 is also a two-character string because it used the answer command to copy the format of Q3.

Range of Answers

The HIGH and LOW commands may be used to specify the highest and lowest acceptable answers. The respondent’s answer is compared to the high and low

values alphabetically, according to the ASCII collating sequence and the extended IBM/PC character set. Upper and lowercase letters are treated equivalently. Both the response and range answers are always capitalized before a comparison is made, regardless of how the UPPERCASE command is set. If a response is outside of a given range of values, the COLLECT and KEYPUNCH programs will display an error message and beep. The interviewer must type an answer that is within this range before he or she can move to another question.

The arguments used with LOW and HIGH commands

- must be delimited by double quotes and
- may contain from one to as many characters as are contained in the answer.

Example

Source Program	Questionnaire Format
.QUESTION = Q1, TYPE = STRING	1. ...
.LOW = "D", HIGH = "H"	...
...	...
...	_ _ _
.ANSWER = 3	LOWEST VALUE = "D"
.NEXT	HIGHEST VALUE = "H"

In this example, the lowest acceptable value has been set to "D," and the highest value is "H." The following example shows how various responses to this question would be evaluated by the COLLECT and KEYPUNCH programs.

Example

Response	In Range?
A	No, too low
APPLE	No, too low
D	Yes
DOG	Yes
FOOT	Yes
H	Yes
HAND	No, too high
ZEBRA	No, too high
?	Not checked ("Don't know" response)

UPPERCASE Command

String questions always use uppercase letters, regardless how the Caps Lock key is set. You can, however, turn this feature off for an individual question, range of questions, or all of the string questions in your questionnaire by using the UPPERCASE

CASE command. When this option is turned off, the interviewer may type using upper or lowercase letters.

When the UPPERCASE command is used within a question, the setting affects only that question. If it is used outside of a question, it affects all of the following questions until another UPPERCASE command is used.

Example

Source Program	Questionnaire Format
.QUESTION = Q1, TYPE = STRING .UPPERCASE = OFFANSWER = 10 .NEXT	1. ... _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
.QUESTION = Q2, TYPE = STRINGANSWER = Q1 .NEXT	2. ... _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
.UPPERCASE = OFF .QUESTION = Q3, TYPE = STRINGANSWER = Q1 .NEXT	3. ... _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
.QUESTION = Q4, TYPE = STRINGANSWER = Q1 .NEXT	4. ... _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
.UPPERCASE = ON	

The UPPERCASE command is used several different ways in this example. It is use within question Q1 to turn its setting off, which allows the interviewer to use upper or lowercase letters. Question Q2 will use uppercase letters because the default setting is still on; the previous setting affected only question Q1. Since the next UPPERCASE command was not used within a question, it will change the default setting for all of the following questions. Here, the settings for both questions Q3 and Q4 will be turned off. The last UPPERCASE command turns the default setting back on.

Open-Ended Answer

Open-ended answer questions are used to enter large amounts of text, which may range from one or two paragraphs to several pages or more in length. This question offers basic word-processing features that make it easy for the interviewer to edit large amounts of text. These include insert and type-over modes, automatic word wrapping, screen scrolling, block copying, block deleting, and word searching.

Usage

```
.IDENTIFY = variable name

.QUESTION = variable name, TYPE = OPENEND
.SUBTITLE = "phrase"
.PUT = variable name
.COPY = variable name
.CLOCK = ON or OFF
.FOREGROUND = color, BACKGROUND = color
.HIGHLIGHT = color, BORDER = color
...
...
...
.ANSWER = buffer size or = variable name
...
.IF (test expression) variable name
.SKIP (target expression)
.SKIPIF (test expression) (target expression)
.NEXT = variable name
```

Notes: Commands and arguments in bold type are required. The other commands are not required, but may be used to modify how the question works. Lowercase words indicate the type of argument a command requires, and ellipses show where text lines should be typed.

Specifying Question Type

The **TYPE** command must be set to "OPENEND."

Answer Field

The open-ended question is different from all the other types of questions because the response that the interviewer types is not stored in the main data file. Instead, it is stored in a separate file that will be put on the same disk and in the same directory as the main data file. This is done to accommodate the variable amount of text that the interviewer may enter.

The field for an open-ended question in the main data file contains only the name of the file that holds the response. The **KEYPUNCH** program, for example, will automatically use this information to retrieve the correct response file when you ask it to print an interview record or create an askSam text-based data file. Likewise, the **CONVERT** program will use the file name when you ask it to put open-ended responses into a generic word processor file format. This allows you to select particular responses and use them with your word-processing program.¹²

¹²You should not load open-ended response files directly into your word processor. They contain formatting codes that are unique to QPL and thus will not be displayed correctly.

Before the COLLECT or KEYPUNCH programs display an open-ended question, they first create a name for the open-ended text file, open the file, and then load it into a window on the screen where the interviewer may type a response. The file name is constructed from three components:

- a case identification number,
- the question variable name, and
- an “OTX” file name extension.

These components allow the QPL programs to generate a unique file name for each open-ended question for each interview record. The COLLECT and KEYPUNCH programs automatically generate a file name and put it into an answer field when the interviewer reaches an OPENEND question.

IDENTIFY Command

The case identification number is copied from another question in your questionnaire. This question must be a 1- to 4-digit integer, and the answer should be entered before an open-ended question is used.¹³ You must use the IDENTIFY command to tell the COMPILE program the name of the question you will be using to hold the unique case identification number.

The variable name of the open-ended question is used for the second half of the file name. It cannot be more than 4 characters long because it will be appended after the identification number in the file name.

The OTX extension is not stored in the data file, but it is used whenever the QPL programs read or write the files.

Example

Source Program	Questionnaire Format
<pre>. IDENTIFY = Q1 .QUESTION = Q1, TYPE = NUMBER Unique case identification code. .ANSWER = 4 .NEXT</pre>	<pre>1. Unique case identification code. _ _ _ _ _ </pre>
<pre>.QUESTION = Q2, TYPE = OPENEND What other comments do you have? .ANSWER .NEXT</pre>	<pre>2. What other comments do you have? _ _ _ _ _ Q 2 - - .OTX</pre>

¹³ A unique case identification number may be entered into a number question in one of three ways: (1) it may be keyed by the interviewer, (2) automatically computed using the COMPUTE command with the COUNT keyword (see chapter 4), or (3) copied from a number question in a control file using the COPY command (see chapter 6).

In this example, question Q1 will hold the case identification number. If the interviewer enters the number 12 for question Q1, for example, the file name for question Q2 will be “0012Q2--.OTX.”

Date and Case Identification Stamp

The open-ended question automatically puts information in the open-ended question response file to identify when information was entered and to which question and interview it belongs. This information includes the

- question number and name,
- case identification number,
- the date and time that the file was first edited, and
- the full text of the question.

This information is only put in the response file the first time it is used.

Example

```
QUESTION: 2 (Q2) Q1=12 7-16-1991 7:38:21 a.m.
```

```
What other comments do  
you have?  
=====
```

```
-
```

This example shows what the response file for question Q2 (case number 12) would contain the first time it is displayed to the interviewer. The editor’s cursor is automatically moved to the bottom of the document, where the interviewer may enter additional text.

Edit Buffer

By default, the open-ended question allows files as large as 4,000 characters to be edited. You can use the ANSWER command, however, to decrease the buffer size to a minimum of 1,000 characters, or increase it to a maximum of 32,000 characters.¹⁴

Example

```
Source Program                               Questionnaire Format  
  
.IDENTIFY = Q1  
  
.QUESTION = Q1, TYPE = NUMBER           1. Unique case ID.  
Unique case ID.  
.ANSWER = 4                               |_|_|_|_|  
.NEXT
```

¹⁴One typed, double-spaced page contains about 2,000 characters.

```
.QUESTION = Q2, TYPE = OPENEND      2. ...  
  ...  
  ...  
.ANSWER                               |__|__|__|__| Q| 2| -| -|.OTX  
.NEXT  
  
.QUESTION = Q3, TYPE = OPENEND      3. ...  
  ...  
  ...  
.ANSWER = 6000                         |__|__|__|__| Q| 3| -| -|.OTX  
.NEXT  
  
.QUESTION = Q4, TYPE = OPENEND      4. ...  
  ...  
  ...  
.ANSWER = Q3                           |__|__|__|__| Q| 4| -| -|.OTX  
.NEXT
```

In this example, the IDENTIFY command is used to tell the COMPILE program that question Q1 is being used to hold a unique case identification number. The number that is entered here will be used as the first part of the file name for all three of the open-ended questions: Q1, Q2, and Q3. The names of these questions will be used for the second half of the names.

The column on the right shows how open-ended questions will be formatted when you use the CONVERT program to print it in questionnaire format. Although only the file name is displayed here, the COLLECT and KEYPUNCH programs will show the name of the file being edited at the bottom of the screen, but display a large window in the middle of the screen where the interviewer may enter text.

Programming Considerations

Although the QPL system of programs does much of the work of keeping track of open-ended response files, there are two situations that you should be concerned about: (1) that there is enough room on an interviewer's floppy disk to store a new interview record and open-ended response files, and (2) that both the main data file (which has a DAT extension) and all of the open-ended response files (which have OTX extensions) are copied when merging the records from different interviewers' disks.

If your questionnaire has many open-ended questions, you must monitor the amount of free disk space an interviewer has available because the COLLECT and KEYPUNCH programs cannot accurately predict when they will run out of room to store a new interview and all of the open-ended response files. The free space on the disk may be used faster than you expect because DOS allocates a minimum of 1,024 bytes to each response file, even if the interviewer has typed fewer characters.

You also may need to be concerned about the number of files that can be stored in the root directory of the floppy disk. The maximum number of files you may put in a root directory varies with the type of disk drive you are using. Generally, the root directory of a double-sided, double-density, 5.25-inch floppy disk can hold up to 112 entries. The root directory of a high density, 3.5-inch floppy disk can hold 224 entries.¹⁵ There is no limit, however, on the number of files you may put in a subdirectory. If you expect to create a large number of files, you should put your program, data, and open-ended response files in a subdirectory on the floppy disk.

Date

There are two types of date questions, and they both provide a reliable way for the interviewer to enter the correct date.¹⁶ The first type of date question, called "DATE," can be used to enter dates from January 1, 1900, to December 31, 1999. The second type of date question, called "LDATE," allows a much larger range of dates to be entered. In this case, dates ranging from January 1, 0001, through December 31, 9999, may be entered in response to a question. Both of these types of questions have several features that help the interviewer enter the correct information.

- Only valid dates may be entered. The interviewer cannot move to another question until a valid date has been entered.
- You may specify an allowable range of dates. If the interviewer enters a date that is out of the range, the COLLECT and KEYPUNCH programs will display an error message and beep. The interviewer cannot move to another question until a date that is within this range has been entered.
- You may use the equation processor's date functions to check for dates that fall on weekends or U.S. federal holidays, or perform other date arithmetic.
- The interviewer can enter a "Don't know" response by pressing the question mark key. A question mark will be displayed on the screen instead of the month.

¹⁵Microsoft *MS-DOS User's Reference*, Microsoft Corporation, p.5, 1987.

¹⁶The date that the interview took place can be automatically recorded in the data file. See the next section on the Current Date.

Usage

```

.QUESTION = variable name, TYPE = DATE or LDATE
.SUBTITLE = "phrase"
.PUT = variable name
.COPY = variable name
.CLOCK = ON or OFF
.FOREGROUND = color, BACKGROUND = color
.HIGHLIGHT = color, BORDER = color
.LOW = (date expression)
.HIGH = (date expression)
.COMPUTE (expression)
.COMPUTEIF (test expression) (expression)
...
...
...
.ANSWER = variable name
...
.IF (test expression) variable name
.SKIP (target expression)
.SKIPIF (test expression) (target expression)
.NEXT = variable name

```

Notes: Commands and arguments in bold type are required. The other commands are not required, but may be used to modify how the question works. Lowercase words indicate the type of argument a command requires, and ellipses show where text lines should be typed.

Specifying Question Type

The TYPE command must be set to “DATE” or “LDATE.” DATE questions are stored in a 6-digit field using YYMMDD format. LDATE (long date) questions are stored in an 8-digit field using YYYYMMDD format.

Example

Source Program	Questionnaire Format
<pre> .QUESTION = Q1, TYPE = DATEANSWER .NEXT </pre>	<pre> 1. 19 _ _ - _ _ - _ _ Year Month Day </pre>
<pre> .QUESTION = Q2, TYPE = LDATEANSWER .NEXT </pre>	<pre> 2. _ _ _ _ _ - _ _ - _ _ Year Month Day </pre>
<pre> .QUESTION = Q3, TYPE = LDATEANSWER 19911009 .NEXT </pre>	<pre> 3. _ _ _ _ _ - _ _ - _ _ Year Month Day </pre>

In this example, question Q1 is a 6-digit date and question Q2 is an 8-digit date. The column on the right shows how these questions would be printed in

questionnaire format by the CONVERT program. The COLLECT and KEYPUNCH programs, however, will display the date to the interviewer in the customary month-day-year format.¹⁷

Default answer lines are not required. The size of the field is always set by the type of date being used. Since a default date was not specified in question Q1, the COLLECT and KEYPUNCH programs will automatically fill the field the current date, which the interviewer may change. For example, if the interviewer was performing the interview on July 4, 1996, the answer to this question would be displayed as 07-04-1996.

Default answers must be written using the format with which the data is stored. In question Q3, the default date for a LDATE question has been written in YYYYMMDD format. When the question is displayed on the screen, the answer line will be shown as “10-09-1991,” which the interviewer may change.

If you do not want a particular date to be displayed, you may set the default answer to all zeros. Then, the interviewer must enter the complete date before he or she will be allowed to go to the next question.

Example

Source Program	Questionnaire Format
.QUESTION = Q1, TYPE = DATEANSWER 000000 .NEXT	1. 19 _ _ - _ _ - _ _ Year Month Day
.QUESTION = Q2, TYPE = LDATEANSWER 00000000 .NEXT	2. _ _ _ _ - _ _ - _ _ Year Month Day
.QUESTION = Q3, TYPE = LDATEANSWER = Q2 .NEXT	3. _ _ _ _ - _ _ - _ _ Year Month Day

In this example, question Q1 initially will be displayed as “00-00-1900” to the interviewer. Question Q2 will be displayed as “00-00-0000.” Question Q3 will also

¹⁷The date may be formatted differently if your computer is configured for a country other than the United States. See chapter 7.

be displayed using zeros because its ANSWER command copies the default answer line from question Q2.

Range of Dates

The earliest date that can be entered in a DATE question is January 1, 1900, and the latest date is December 31, 1999. LDATE questions have an even wider range of dates that may be entered. You can use the LOW and HIGH commands, however, to limit the range of dates that the interviewer may enter. You may set the range to a constant date, the date given in the answer to a previous question, or a date calculated in an expression.

The value of a 6-digit DATE question is always promoted to an 8-digit LDATE question before it is used in a LOW, HIGH, or COMPUTE command. That is, even though the date July 4, 1996, is stored as 960704 by a DATE question, it will be treated by a LOW, HIGH, or COMPUTE command as if it were 19960704. This feature makes it easier for you to compare DATE and LDATE values, since you do not need to worry about the assumed “19” century. When comparing a DATE or LDATE value to a constant date, always write the constant date in the 8-digit (YYYYMMDD) format.

Example

Source Program	Questionnaire Format
.QUESTION = Q1, TYPE = DATE .LOW = 19890101, HIGH = 19891231ANSWER .NEXT	1. 19 _ _ - _ _ - _ _ Year Month Day LOWEST VALUE = 19890101 HIGHEST VALUE = 19891231
.QUESTION = Q2, TYPE = DATE .HIGH = Q1ANSWER .NEXT	2. 19 _ _ - _ _ - _ _ Year Month Day HIGHEST VALUE = #1
.QUESTION = Q3, TYPE = LDATE .LOW = Q1ANSWER .NEXT	3. _ _ _ _ - _ _ - _ _ Year Month Day LOWEST VALUE = #1

In this example, question Q1 is a 6-digit DATE question that has limits on both the earliest and latest dates that may be entered by the interviewer. In this case, the interviewer must enter a date between January 1 and December 31, 1989, inclusive. If the interview enters a “Don’t know” response (by pressing the question mark key), a range check is not made.

Question Q2 also is a DATE question. It has a limit only on the latest date that may be entered. This date may be no later than the date that was entered for question Q1.

Question Q3, however, is an 8-digit date. The comparison to question Q1 works correctly because its value is promoted to an 8-digit date before it is compared with the value of question Q3.

There are several important points that you should keep in mind when specifying ranges that are based on the answers to other questions.

- Refer only to questions that have already been answered.
- Both questions must be on the same path (*i.e.*, the interviewer should not be able to skip around the first question).
- You should not specify a high and low value from the same question.
- If the interviewer entered a “Don’t know” value for either question, the range check is not made.

Date Functions

The QPL equation processor has a number of functions that you can use to perform date arithmetic. These functions allow you to compute the number of days between dates, add days to a date, find the day of the week a date falls on, and determine whether a date falls on a weekend or U.S. federal holiday.

Date Functions

TOJUL()	Return Julian date from Gregorian date
FROMJUL()	Return Gregorian date from Julian date
JDAYOFWEEK()	Return a code that indicates the day of week from a Julian date
JFHOLIDAY()	Return a code that indicates a weekend or U.S. federal holiday from a Julian date
JYEAR()	Return year from Julian date
JMONTH()	Return month from Julian date
JDAY()	Return day from Julian date
GYEAR()	Return year from Gregorian date
GMONTH()	Return month from Gregorian date
GDAY()	Return day from Gregorian date

Note: Examples using each function are given in chapter 4.

The QPL equation processor uses Julian dates to perform date arithmetic. Julian dates differ from normal, or Gregorian, dates by using one integer to represent a particular day instead of three. When a date is converted to its Julian equivalent, it is easy to add or subtract days, or find the number of days between two Julian dates.

While many software packages use something similar to Julian dates for this purpose (such as the Lotus @DATE function), the QPL equation processor uses the same Julian dates that are used in astronomy, with the one variation that the Julian days used here begin at midnight instead of noon.¹⁸ The Julian date functions used here consider leap years and the 10-day gap that occurred in October 1582, when the Julian calendar was converted to the Gregorian calendar.

The following example shows how you could use the date functions to make sure that the interviewer cannot enter a date that falls on a weekend or U.S. federal holiday.

Example

Source Program	Questionnaire Format
.QUESTION = Q1, TYPE = DATEANSWER .IF (Q1 = MISSING) Q3 .IF (JFHOLIDAY(TOJUL(Q1))) Q2 .NEXT = Q3	1. 19 _ _ - _ _ - _ _ Year Month Day IF (#1 = MISSING) GO TO #3 IF (JFHOLIDAY(TOJUL(#1))) GO TO #2 SKIP TO QUESTION 3 =====
.QUESTION = Q2, TYPE = VOID The date you entered falls on a weekend or federal holiday. Please enter a new date. .ANSWER Press PgUp to make correction... .NEXT	2. The date you entered falls on a weekend or federal holiday. Please enter a new date. Press PgUp to make correction...
.QUESTION = Q3...	3. ...

¹⁸The QPL Julian date functions are based on the C functions by Flanery, Teukolsky, and Vettering, *Numerical Recipes in C*, pp. 10-13, and Clark, "A Date Object in C++," *The C Users Journal*, pp. 57-69, June 1990.

After a date has been entered, the first IF command will be evaluated to determine if the interviewer entered a “Don’t know” response by pressing the question mark key. If the date is not known, the COLLECT and KEYPUNCH programs will skip to question Q3. Otherwise, the next IF command will be evaluated.

The second IF command checks to see if the date falls on a weekend or holiday. The date entered in question Q1 first is converted to a Julian date, and then the Julian date is converted to a number that indicates whether the date is a weekend or holiday.¹⁹ If this function returns a value greater than zero, then the date is a weekend or holiday, and the COLLECT and KEYPUNCH programs will skip to question Q2. If it returns a value of zero, the expression will be false, and the NEXT command will be used to skip to question Q3.

Question Q2 is used as an error message to the interviewer. Since it is a VOID question, no data will be stored in the data file. It tells the interviewer to press the PgUp key to back up and correct the date.

Computing Responses

The COMPUTE and COMPUTEIF commands may be used with DATE and LDATE questions to automatically calculate response dates. This could be useful, for example, if you need to determine the date on which an event should have taken place. The following example uses compute commands to determine when a decision should have been made on an application.

Example

Source Program

```
.QUESTION = Q1, TYPE = DATE
When did you submit
your application?
.ANSWER
.NEXT

.QUESTION = Q2, TYPE = DATE
Calculate latest decision
date allowed.
.ANSWER
.COMPUTEIF (NOT(JFHOLIDAY(TOJUL(Q1) + 30))) (FROMJUL(TOJUL(Q1) + 30))
.COMPUTEIF (NOT(JFHOLIDAY(TOJUL(Q1) + 31))) (FROMJUL(TOJUL(Q1) + 31))
.COMPUTEIF (NOT(JFHOLIDAY(TOJUL(Q1) + 32))) (FROMJUL(TOJUL(Q1) + 32))
.COMPUTE (FROMJUL(TOJUL(Q1) + 33))
.NEXT
```

¹⁹See chapter 4 for a list of return codes for the JFHOLIDAY() function.

Chapter 3 Types of Questions

```
.QUESTION = Q3, TYPE = MULT
Was a decision on your application
made by [Q2]?
.ANSWER
Yes
No
Don't know
.NEXT
```

Questionnaire Format

1. When did you submit
your application?

```
19|_|_|-|_|_|-|_|_|
   Year   Month   Day
```

2. Calculate latest decision
date allowed.

```
19|_|_|-|_|_|-|_|_|
   Year   Month   Day
```

```
COMPUTEIF (NOT(JFHOLIDAY(TOJUL(#1) + 30))) (FROMJUL(TOJUL(#1) + 30))
COMPUTEIF (NOT(JFHOLIDAY(TOJUL(#1) + 31))) (FROMJUL(TOJUL(#1) + 31))
COMPUTEIF (NOT(JFHOLIDAY(TOJUL(#1) + 32))) (FROMJUL(TOJUL(#1) + 32))
COMPUTE (FROMJUL(TOJUL(#1) + 33))
```

3. Was a decision on your application
made by #2?

(CHECK ONLY ONE ANSWER)

```
|_| 1. Yes
|_| 2. No
|_| 3. Don't know
```

For this example, a decision must be taken on an application within 30 days of its submission. If the last day a decision may be made falls on a weekend or U.S. federal holiday, the decision must be made on the next business day. The first COMPUTEIF command in question Q2 checks to see if a date 30 days after the submission date is a weekend or holiday. If it is not, then that date is used as a response for question Q2. If it is a weekend or holiday, then the next day is checked, and so on.

Finally, the date that a decision should have been made is displayed in the text to question Q3, which asks the respondent if a decision was made by that date. For example, if the respondent answered June 4, 1991, to question Q1, the date July 5, 1991, will be displayed. These dates are 31 days apart because the 30th day was July 4, 1991, a U.S. federal holiday.

Current Date

You should always include a current date question in your questionnaires. During analysis, it is often useful to know when an interview took place, especially when tracking down data-entry errors. The COLLECT and KEYPUNCH programs read the date from the computer's internal clock and write it to each record in YYMMDD format. Thus, it is important that the interviewers be trained to correctly set the date on the computer before starting the COLLECT or KEYPUNCH programs.²⁰

The interviewer never sees this question on the COLLECT screen. It simply inserts the current date into the record and moves on to the next question. The KEYPUNCH program displays the question on the screen, but it does not allow the interviewer to change the answer. It inserts the current date and moves to the next question. Once the date is set, it cannot be changed later if the record is reloaded. When printed in questionnaire format (using the CONVERT program), however, it is shown as a normal date question.

Usage

```
.QUESTION = variable name, TYPE = XDATE  
.SUBTITLE = "phrase"  
.PUT = variable name  
...  
...  
...  
.ANSWER  
...  
.IF (test expression) variable name  
.SKIP (target expression)  
.SKIPIF (test expression) (target expression)  
.NEXT = variable name
```

Notes: Commands and arguments in bold type are required. The other commands are not required, but may be used to modify how the question works. Lowercase words indicate the type of argument a command requires, and ellipses show where text lines should be typed.

Specifying Question Type

The TYPE command must be set to "XDATE."

Question Text Lines

Although this question is not displayed on the screen by the COLLECT program, at least one question text line is required. This line will be displayed by the KEYPUNCH program, printed in questionnaire format by the CONVERT program, and used as a variable label by the CONVERT program when it creates SPSS and SAS programs.

²⁰You may use a DOS batch file program to start the COLLECT and KEYPUNCH programs that ask the interviewer to enter the date and time if they are not set correctly. Example batch files are shown in chapters 9 and 10.

Field Size

The field size will always be set to six characters. A default answer is not required because the date will always be set to the current date at the time of the interview.

Range of Dates

The value of a current date question can be used by a DATE or LDATE question to limit the range of dates that the interviewer may enter. For example, if a question asked for a date that had to occur before the date of the interview, you could limit the latest date by referring to the current date question with the HIGH command.

Example

Source Program	Questionnaire Format
.QUESTION = Q1, TYPE = XDATE	1. ...
...	...
.ANSWER	19 _ _ - _ _ - _ _
.NEXT	Year Month Day
.QUESTION = Q2, TYPE = DATE	2. ...
.HIGH = Q1	...
...	...
.ANSWER	19 _ _ - _ _ - _ _
.NEXT	Year Month Day
	HIGHEST VALUE = #1

In this example, question Q1 automatically records the date of the interview. Question Q2 compares the date that is entered by the interviewer to the date that was recorded in Q1. If the interviewer enters a date that is later than Q1, the COLLECT and KEYPUNCH programs will beep and give the interviewer an error message. The interviewer must enter a date that is the same or earlier than Q1 before he or she may move to the next question. If the interviewer enters a “Don’t know” response, by pressing the question mark key, the range check is not made.

Current Time

The time of day that the interview took place can be automatically recorded by the COLLECT and KEYPUNCH programs. The interviewer never sees this question on the COLLECT screen. It is, however, displayed by the KEYPUNCH program, although the interviewer may not change it. The COLLECT and KEYPUNCH programs read the time from the computer's built-in clock and insert it into the record, and then move to the next question. Thus, interviewers must be trained to correctly set the time on the computer before starting the COLLECT and KEYPUNCH programs.²¹

This question can also be used to measure how long the respondent took to answer a question. For example, it could be used to measure how long the interviewer was placed on hold by the respondent, or how long it took to complete the entire interview. Any number of points can be timed in a questionnaire.

The interviewer may set the time only on the first pass through the question. If the interviewer backs up and then goes through the question again, the time is not changed.

Usage

```
.QUESTION = variable name, TYPE = TIME  
.SUBTITLE = "phrase"  
.PUT = variable name  
.COMPUTE (expression)  
.COMPUTEIF (test expression) (expression)  
...  
...  
...  
.ANSWER  
.IF (test expression) variable name  
.SKIP (target expression)  
.SKIPIF (test expression) (target expression)  
.NEXT = variable name
```

Notes: Commands and arguments in bold type are required. The other commands are not required, but may be used to modify how the question works. Lowercase words indicate the type of argument a command requires, and ellipses show where text lines should be typed.

Specifying Question Type

The TYPE command must be set to "TIME."

Question Text Lines

Although this question is not displayed on the screen by the COLLECT program, at least one question text line is required. This line will be displayed by the KEYPUNCH program, printed in questionnaire format by the CONVERT program, and used as a variable label by the by the CONVERT program.

²¹You may use a DOS batch file program to start the COLLECT and KEYPUNCH programs that ask the interviewer to enter the date and time if they are not set correctly. Example batch files are shown in chapters 9 and 10.

Field Width and Output

The time is always stored internally as the number of seconds that have elapsed since midnight, in a five-digit field. Since the computer fills in the time when the interviewer passes through this question in the interview, no default answer line is required.

Example

Time	Data File
12:01 a.m.	00060
8:00 a.m.	28800
8:30 a.m.	30600
12:00 noon	43200
3:30 p.m.	55800
6:15 p.m.	65700

The time can be returned to hour-and-minute format, for example, using the following formulas when analyzing the results using SPSS:

SPSS-X Example

```
COMPUTE HR = TRUNC(T1 / 3600) /* Hour
COMPUTE MIN = ((T1 / 3600) - HR) * 60 /* Minute
```

Note: TRUNC is an SPSS function that converts values to integers by discarding the remainder; T1 is the variable name of a time question; 3600 is the number of seconds per hour; and 60 is the number of minutes per hour.

This example converts the time from seconds-since-midnight to standard military time in hours and minutes.

Timing a Question

The length of time a respondent takes to answer a question, or series of questions, can be measured by using two or more time questions.²² The elapsed time may be calculated later during analysis of the questionnaire by subtracting the starting time from the ending time.

Time questions, which are not seen by the interviewer, should be placed before and after a question, or group of questions. The starting time will be set by the COLLECT or KEYPUNCH program when the interviewer passes through the first time question, and the ending time will be set after he or she passes through the last time question.

²²See also the CLOCK command, which was described in the multiple choice question section of this chapter.

Example

Source Program	Questionnaire Format
.QUESTION = T1, TYPE = TIME Start time .ANSWER .NEXT	1. Start time _ _ _ _ _ _ _
.QUESTION = Q1, TYPE = MULT Press ENTER when someone answers the phone. .ANSWER Phone answered .NEXT	2. Press ENTER when some answers the phone. _ 1. Phone answered
.QUESTION = T2, TYPE = TIME End time .ANSWER .NEXT	3. End time _ _ _ _ _ _ _

In this example, the starting time (question T1) will be set before the interviewer sees question Q1. After the interviewer hears that phone has been answered, he or she is instructed to press the Enter key to move to the second time question (T2). The COLLECT and KEYPUNCH programs will automatically record the ending time, and move to the following question.

The time it took the interviewer to go from question T1 to T2 (*i.e.*, the time it took to answer question Q1) can be calculated by subtracting the value of T2 from T1.

SPSS-X Example

```
COMPUTE NETSEC = T2 - T1          /* Net Seconds  
COMPUTE NETMIN = (T2 - T1) / 60  /* Net Minutes  
COMPUTE NETHR  = (T2 - T1) / 3600 /* Net Hours
```

The above example shows how the elapsed time would be calculated when analyzing the results using SPSS. The elapsed time can also be converted into minutes or hours by dividing by 60 seconds per minute or 3,600 seconds per hour.

You also may base skip instructions in your questionnaire on this same calculation. An IF command could be added to question T2 to make the COLLECT and KEYPUNCH programs skip to different questions according to how much time was taken to answer Q1.

Chapter 3 Types of Questions

Example

Source Program	Questionnaire Format
.QUESTION = T2, TYPE = TIME	3. End time
End time	
.ANSWER	_ _ _ _ _ _ _
.IF (T2 - T1 > 120) Q3	
.NEXT	IF (#3 - #1 > 120) GO TO #5

Question T2 has been modified in this example so that it will skip to question Q3 if it took longer than two minutes (120 seconds) to answer question Q1. If it took less time, the next question in the list will be asked since no other skip instructions have been used.

Computing Responses

In addition to automatically recording the time when the interviewer first passes through a TIME question, you may use the COMPUTE and COMPUTEIF commands to calculate other times and display them in hour-and-minute format in the text of other questions. This may be useful when you need to ask a respondent if he or she performed some action by a particular time.

Example

```
Source Program

.QUESTION = Q1
.TYPE = NUMBER = "##.## (hours.minutes)"
What time did you arrive?
(Enter response using military time.)
.ANSWER
.NEXT

.QUESTION = Q2, TYPE = TIME
Compute arrival time in seconds.
.ANSWER
.COMPUTE (FLOOR(Q1)*3600 + MOD(Q1)*100*60)
.NEXT

.QUESTION = Q3, TYPE = TIME
Compute latest allowable departure time.
.ANSWER
.COMPUTE (Q2 + 2 * 3600)
.NEXT

.QUESTION = Q4, TYPE = MULT
Did you depart by [Q3]?
.ANSWER
Yes
No
Don't know
.NEXT
```

Chapter 3
Types of Questions

Questionnaire Format

1. What time did you arrive?

(Enter response using military time.)

|_|_|.|_|_| (hours.minutes)

2. Compute arrival time in seconds.

|_|_|_|_|_|_|

COMPUTE (FLOOR(#1) * 3600 + MOD(#1) * 100 * 60)

3. Compute latest allowable departure time.

|_|_|_|_|_|_|

COMPUTE (#2 + 2 * 3600)

4. Did you depart by #3?

(CHECK ONLY ONE ANSWER)

|_| 1. Yes
|_| 2. No
|_| 3. Don't know

Notes: The FLOOR() function returns the largest integer that is less than or equal to a real number. The MOD() function returns the decimal portion of a real number. See chapter 4, for more information on QPL equation processor functions.

In this example, a COMPUTE command is used in question Q2 to convert the time in question Q1 to the number of seconds that have elapsed since midnight. Question Q3 computes the time two hours later. Finally, question Q4 asks the respondent if he or she had departed by this time. For example, if the respondent said that he or she arrived at 2:30 p.m. (14.30 hours), question Q4 would be displayed as “Did you depart by 4:30 p.m.?”

TIME Keyword

You can make a TIME question record the current time whenever the interviewer passes through it by using the COMPUTE command and the TIME keyword. The TIME keyword is used in QPL expressions, and it always returns the current time in seconds.

Example

Source Program	Questionnaire Format
.QUESTION = Q1, TYPE = TIMEANSWER .NEXT	1. ... _ _ _ _ _ _ _
.QUESTION = Q2, TYPE = TIMEANSWER .COMPUTE (TIME) .NEXT	2. ... _ _ _ _ _ _ _ COMPUTE TIME
.QUESTION = Q3, TYPE = VOID The first time is: [Q1]. The second time is: [Q2]. .ANSWER Press ENTER to continue... .NEXT	3. The first time is: #1. The second time is: #2. Press ENTER to continue...

In this example, question Q1 is a normal TIME question. When the interviewer passes through it, the time will be recorded. It will not be changed if the interviewer backs up to it again or reloads the record using the KEYPUNCH program. The time at question Q2, however, will be set every time the interviewer passes through the question. Question Q3 will display the answers to questions Q1 and Q2 using hour-and-minute format.

Version Number

A version number can be put in a field on each record using this question type. The COLLECT and KEYPUNCH programs automatically enter the version number on each record as the interviewer passes through the question. The question itself, however, will not be displayed on the screen.

Including version numbers in the questionnaire can be useful, for example, if the questionnaire may be changed as it is being used.²³ The version number can be used in analysis to detect any changes in the responses, based on the new wording.

If your survey will be using more than one floppy disk, the version number could be used to identify the disk on which a record originally was written. This information may be useful later if problems arise after data on separate disks have

²³If your changes alter the record layout, by adding or dropping questions or changing the size of a question field, you must use the CONVERT program to translate your old data to the layout of the revised questionnaire program. See chapter 11.

been merged together, such as accidentally appending one data set more than once.

Usage

```
.QUESTION = variable name, TYPE = VERSION  
.SUBTITLE = "phrase"  
.PUT = variable name  
...  
...  
...  
.ANSWER  
...  
.IF (test expression) variable name  
.SKIP (target expression)  
.SKIPIF (test expression) (target expression)  
.NEXT = variable name
```

Notes: Commands and arguments in bold type are required. The other commands are not required, but may be used to modify how the question works. Lowercase words indicate the type of argument a command requires, and ellipses show where text lines should be typed.

Specifying Question Type

The TYPE command must be set to “VERSION.”

Field Size

The field size for version questions is always set by the default answer text line. Only one answer line may be used, but it may contain any combination of letters or numbers. A carriage return must be typed immediately after the last character in the answer line. Whatever is written in the answer line will always be copied to each record, without any changes.

Example

```
Source Program                               Questionnaire Format  
  
.QUESTION = Q1, TYPE = VERSION              1. Version number, Disk number  
Version number, Disk number  
.ANSWER                                     | 1 | 3 |  
13  
.NEXT
```

In this example, question Q1 is a version question with a default answer of 13. In this case, the “1” refers to the version number of the questionnaire, and the “3” refers to the disk number. Separate version questions could also have been used to define these numbers. The COLLECT and KEYPUNCH programs always will write “13” to each record.

Void

There are often times when you may want to display information to the interviewer, but do not need to have the interviewer record a response. For example, if the interviewer is only reading instructions to the respondent before asking the first question, you could put that information in a void question type. The void question type is displayed on the screen by the COLLECT and KEYPUNCH programs just like any other question, but it does not record any responses in the data file.

Usage

```
.QUESTION = variable name, TYPE = VOID
.SUBTITLE = "phrase"
.CLOCK = ON or OFF
.FOREGROUND = color, BACKGROUND = color
.HIGHLIGHT = color, BORDER = color
...
...
...
.ANSWER
...
.IF (test expression) variable name
.SKIP (target expression)
.SKIPIF (test expression) (target expression)
.NEXT = variable name
```

Notes: Commands and arguments in bold type are required. The other commands are not required, but may be used to modify how the question works. Lowercase words indicate the type of argument a command requires, and ellipses show where text lines should be typed.

Specifying Question Type

The TYPE command must be set to "VOID."

Field Size

Void questions never save data in the record; thus, they always have a field size of zero. You must, however, include one answer text line. This is usually just a short message to the interviewer such as, "Press ENTER to begin..."

Example

Source Program	Questionnaire Format
.QUESTION = Q1, TYPE = VOID	1. ...
...	...
...	
.ANSWER	Press ENTER to begin...
Press ENTER to begin...	
.NEXT	

Random Number

A random number question is generally used to randomize the order in which a group of questions are asked. This may be important if you believe that the order may affect how a respondent may answer a particular question.

A random number question is programmed like a multiple choice question, except that you must specify the **RANDOM** keyword at the **TYPE** command instead of **MULT**. The **COLLECT** and **KEYPUNCH** programs will randomly select an answer from the list when the interviewer passes through the question. Once an answer has been selected, the interviewer cannot change the selection by backing up and passing through the question again.²⁴ The **COLLECT** program does not display the question on the screen. Instead, it will immediately skip to the question specified by any **GOTO**, **IF**, **SKIP**, **SKIPIF**, and **NEXT** commands that were used in the random number question.

It is usually a good idea to change your random question to a **MULT** question while you are developing your program. This makes it easier to test each of the paths that follow from it (since the question will be displayed on the screen and you can back up and change the answer). After you are sure that the skip instructions are working properly, you can change the type back to a **RANDOM** question.

Usage

```
.QUESTION = variable name, TYPE = RANDOM  
.SUBTITLE = "phrase"  
.PUT = variable name  
...  
...  
...  
.ANSWER = variable name  
...  
.GOTO = variable name  
...  
...  
.IF (test expression) variable name  
.SKIP (target expression)  
.SKIPIF (test expression) (target expression)  
.NEXT = variable name
```

Notes: Commands and arguments in bold type are required. The other commands are not required, but may be used to modify how the question works. Lowercase words indicate the type of argument a command requires, and ellipses show where text lines should be typed.

Specifying Question Type

The **TYPE** command must be set to “**RANDOM**.”

²⁴A random number question will have a default value of zero before the interview passes through it.

Answer Lines

At least one answer line is required. Since this question is not displayed on the screen by the COLLECT program, there is essentially no limit on the number of answer lines you may have.²⁵

Skip Instructions

The GOTO, IF, SKIP, SKIPIF, and NEXT commands can be used to skip to particular questions according to what response is randomly chosen. The following example demonstrates how a random number question could be used to randomize the order in which three questions are asked. In this example, questions Q1, Q2, and Q3 will always be asked in the same circular order, but question R1 will determine which question will be asked first.

Example

Source Program	Questionnaire Format
<pre>.QUESTION = R1, TYPE = RANDOM Randomize starting question. .ANSWER Skip to first question .GOTO = Q1 Skip to second question .GOTO = Q2 Skip to third question .GOTO = Q3 .NEXT = Q4</pre>	<pre>1. Randomize starting question. (CHECK ONLY ONE ANSWER) __ 1. Skip to first question GO TO QUESTION 2 __ 2. Skip to second question GO TO QUESTION 3 __ 3. Skip to third question GO TO QUESTION 4 SKIP TO QUESTION 5 =====</pre>
<pre>.QUESTION = Q1, TYPE = MULT First question. .ANSWERIF (R1 = 2) Q4 .NEXT</pre>	<pre>2. First question. (CHECK ONLY ONE ANSWER) __ 1. ... IF (#1 = 2) GO TO QUESTION 4</pre>
<pre>.QUESTION = Q2, TYPE = MULT Second question. .ANSWERIF (R1 = 3) Q4 .NEXT</pre>	<pre>3. Second question. (CHECK ONLY ONE ANSWER) __ 1. ... IF (#1 = 3) GO TO QUESTION 4</pre>

²⁵Maximum is 32,000 lines.

```
.QUESTION = Q3, TYPE = MULT      4. Third question.
Third question.                  (CHECK ONLY ONE ANSWER)
.ANSWER                          |__| 1. ...
      ...                        IF (#1 = 1) GOTO QUESTION 4
.AIF (R1 = 1) Q4                 SKIP TO QUESTION 1
.NEXT = Q1                        =====

.QUESTION = Q4, TYPE = VOID      5. This is the first question
This is the first question       after the randomized loop
after the randomized loop        of questions.
of questions.                   Press ENTER to continue...
.ANSWER                          Press ENTER to continue...
Press ENTER to continue...
.NEXT
```

In this example, question R1 has three possible responses. Each answer corresponds to a question in the circular loop. When the interviewer passes through this question, the COLLECT and KEYPUNCH programs will randomly pick one of the answers. The GOTO commands tell these programs to skip to the question that corresponds to the answer that was randomly selected. This sets the starting question. Once the interviewer has passed through question R1, he or she cannot back up and change the answer (*i.e.*, the random starting point). Since the answer to R1 is stored in the data file, you can use it later during analysis to determine if the order of questions affected the responses.

The NEXT command in question R1 must be set to jump over the circular list of questions. This is important because a random question has a default value of **zero** until the interviewer passes through it. If you had omitted this instruction, the default pathway would run through the random question and fall into the circular loop at question Q1 because none of the GOTO instructions would be true. Since all of the IF statements in the circular loop of questions would be false, the pathway would be an endless loop.

This becomes a problem when the interviewer presses the Escape key to exit a questionnaire early. After the interviewer answers the last question, the COLLECT and KEYPUNCH programs will evaluate the skip path from the beginning to the end of the questionnaire. If they find that the path loops back upon itself (that is, it is caught in an endless loop and cannot get to the last question), it will display an error message saying that a logic error occurred. If this happens, the data will not be stored in the main data file. Instead, a file called LOGIC.ERR will be

created that will list the values of each question and show where the logic error occurred.

Setting the NEXT command to the first question after the circular loop, question Q4 in this case, prevents this problem. If the interviewer presses the Escape key before an answer to question R1 has been set, the NEXT command will send the default path from R1 to Q4 because none of the GOTO commands were true.

The IF commands in questions Q1, Q2, and Q3 are used to exit the loop after all the questions have been asked. These IF commands may be interpreted as saying “If the loop was started at the next question, then exit the loop now.” If this is not true, then the next question should be asked. The NEXT command at question Q3 must be set to the first question, Q1, in order to complete the loop.

Try picking different answers for question R1 and tracing through the GOTO and IF statements to assure yourself that this logic works.

SKIPIF Command

The three GOTO commands in the previous example could be replaced by a single SKIPIF command since the questions in the loop are in consecutive order. The SKIPIF command is convenient to use when you will be using the RANDOM question to skip to one of a large number of questions in a loop.

Example

Source Program	Questionnaire Format
.QUESTION = R1, TYPE = RANDOM	1. Randomize starting question.
Randomize starting question.	
.ANSWER	(CHECK ONLY ONE ANSWER)
Skip to first question	
Skip to second question	__ 1. Skip to first question
Skip to third question	__ 2. Skip to second question
.SKIPIF (R1 > 0) (!Q1 + R1 - 1)	__ 3. Skip to third question
.NEXT = Q4	
	SKIP IF (#1 > 0) (2 + #1 - 1)
	SKIP TO QUESTION 5
	=====

This example shows how question R1 could be rewritten using the SKIPIF command. Note that it explicitly checks the value of R1 before it computes the number of the question it should skip to. If R1 is zero (*i.e.*, a random number has not been selected yet), then the SKIPIF command will be false, and the NEXT command will be used to skip to question Q4. See chapter 4 for more information on the SKIP and SKIPIF commands.

Skip Instructions

The order in which questions are asked can be controlled in a QPL questionnaire using several types of skip commands. These commands allow very complicated questionnaires to be developed. They can be used to ask respondents questions that are appropriate to their circumstances, and skip over questions that are not appropriate.

The COLLECT and KEYPUNCH programs rigidly enforce skip instructions. They do not allow the interviewer to jump around in the questionnaire. He or she only may proceed, question by question, forward or backward, along a single path determined by the skip commands. Although this may seem restrictive, it is a critical feature of the QPL system that insures the integrity of the skip logic.

The path always begins with the first question in your questionnaire program and ends at the last question. In between these points, you can create any number of paths that go in any direction through your questionnaire. When the interviewer answers the last question, the COLLECT and KEYPUNCH programs will find the path the interviewer took and set all of the answers that were not on this path to blank spaces (*i.e.*, the missing value code) before saving the record to the data file. Thus, if an interviewer proceeded down a certain path, but then backed up and proceeded down another path, all of the answers that were entered on the old path will be erased before the record is added to the data file.

The CONVERT program will include skip instructions when you create a formatted version of your questionnaire program. This document will refer to each question by its number instead of its variable name.

Unconditional Skips

Two commands, NEXT and SKIP, may be used to tell the COLLECT and KEYPUNCH programs to perform an unconditional skip after a question has been answered. These skips are called unconditional because they will always be executed, unless they are superseded by a conditional skip instruction.

NEXT Command

The NEXT command, which defines the end of a question, also can be used to specify which question should be displayed after its question is answered. If a particular question is not specified, the COLLECT and KEYPUNCH programs will display the next question in the list. An unconditional skip can be used at the end of any type of question.

Example

Source Program	Questionnaire Format
.QUESTION = Q1, TYPE = STRINGANSWER = 5 .NEXT = Q7	1. _ _ _ _ _ PLEASE SKIP TO QUESTION 7 =====

In this example, the NEXT command has been set to jump to question Q7, wherever it may be in the questionnaire. The COLLECT and KEYPUNCH programs will automatically jump to question Q7 after this question has been answered. The column on the right shows how the CONVERT program will write these skip instructions.

SKIP Command

The SKIP command is less commonly used than the NEXT command. Instead of only skipping to one question, like the NEXT command, you may program the SKIP command to jump to one question in a series. This command is typically used to jump to a random starting point in a circular series of questions.

Example

Source Program	Questionnaire Format
.QUESTION = R1, TYPE = RANDOM Pick random starting question. .ANSWER First Second Third Fourth .IF (R1 = 0) FINISH .SKIP (!Q1 + R1 - 1) .NEXT	1. Pick random starting question. _ 1. First _ 2. Second _ 3. Third _ 4. Fourth IF (#1 = 0) GO TO QUESTION 6 SKIP (2 + #1 - 1)
.QUESTION = Q1, TYPE = MULTANSWER Yes No Don't know .IF (R1 = 2) FINISH .NEXT	2. ... _ 1. Yes _ 2. No _ 3. Don't know IF (#1 = 2) GO TO QUESTION 6

Chapter 4

Skip Instructions

```
.QUESTION = Q2, TYPE = MULT      3. ...
...
.ANSWER = Q1                      |__| 1. Yes
.IF (R1 = 3) FINISH                |__| 2. No
.NEXT                              |__| 3. Don't know

                                IF (#1 = 3) GO TO QUESTION 6

.QUESTION = Q3, TYPE = MULT      4. ...
...
.ANSWER = Q1                      |__| 1. Yes
.IF (R1 = 4) FINISH                |__| 2. No
.NEXT                              |__| 3. Don't know

                                IF (#1 = 4) GO TO QUESTION 6

.QUESTION = Q4, TYPE = MULT      5. ...
...
.ANSWER = Q1                      |__| 1. Yes
.IF (R1 = 1) FINISH                |__| 2. No
.NEXT = Q1                         |__| 3. Don't know

                                IF (#1 = 1) FINISH

                                SKIP TO QUESTION 1
                                =====

.QUESTION = FINISH, TYPE = VOID  6. ...
...
.ANSWER                            Press ENTER to finish.
Press ENTER to finish.
.NEXT
```

The SKIP command works by calculating the number of the question where it will skip. In this example, the expression used at the SKIP command will evaluate to the number of one of four questions: Q1, Q2, Q3, or Q4. After the COLLECT or KEYPUNCH programs have found the question number, they will skip to that question.

The expression first finds the number of the first question in the series and then adds the random number from question R1 to offset the starting point. The exclamation mark (!) tells the COMPILE program to use the number of a question as a value in the expression instead of a variable. Finally, 1 must be subtracted from the total because the random number question codes its answer beginning with 1 instead of zero. For example, if the random number question was set to 3, the expression would be “2 + 3 - 1,” which evaluates to 4, which is the number of the third question in the series Q1-Q4.

The IF commands in question Q1 to Q4 are used to exit from the circular loop of questions. Each of the IF commands checks to see if the next question has already been asked, according to the random number question, R1. If it has, then

the loop will be exited. The NEXT command at question Q4 is set to jump to the top of the list to complete the loop.

Since the random number question will have a value of zero before it is displayed by the COLLECT or KEYPUNCH programs, an IF command must be used at question R1 to move the default path over the circular list of questions. Otherwise, the default path would go into the circular loop, but it could not get out because none of the following IF commands will be true. If the interviewer exited the questionnaire early, by pressing the Escape key, the COLLECT and KEYPUNCH programs would not be able to find a unique pathway from the beginning to the end of the questionnaire and an error message would be displayed. Checking for a zero value before performing the skip prevents this problem. When the interviewer passes through the random number question, it will be set to a value between one and four, making it possible to exit the loop.

The SKIP command can be used, as in the above example, to replace GOTO or IF commands. Using it is particularly convenient when you need to branch to one of a large number of questions. It is less flexible than the GOTO and IF commands, however, because it requires that the series of questions be contiguous so that a simple expression can be written to find the correct question number.

Conditional Skips

There are several commands that can be used to write conditional skip instructions. These commands are conditional because they depend on how the questionnaire is answered. The COLLECT and KEYPUNCH programs will skip to a question only if the condition you specified exists. GOTO commands are used to link skip instructions to particular answers in questions that have a closed list of answers. IF and SKIPIF commands may be used with any type of question to perform skips based on more complex criteria.

GOTO Command

A GOTO command can be linked to any answer in a multiple choice, check-all-that-apply, or random number question to specify which question should be displayed next if that answer is chosen. GOTO statements are optional. If the interviewer chooses an answer that does not have a GOTO command linked to it, the COLLECT and KEYPUNCH programs will skip to the question specified by the NEXT command.

Example

Source Program	Questionnaire Format
.QUESTION = Q1, TYPE = MULT	1. ...
...	...
...	...
...	...
.ANSWER	(CHECK ONLY ONE ANSWER)
Yes	
.GOTO = Q6	__ 1. Yes (GO TO QUESTION 6)
No	__ 2. No
Don't know	__ 3. Don't know
.NEXT = Q33	
	PLEASE SKIP TO QUESTION 33
	=====

In the above example, if the interviewer selects the answer “Yes,” the COLLECT and KEYPUNCH programs will skip to question Q6. If another answer is selected, they will jump to the question specified in the NEXT command. In this case, they will jump to question Q33. If the NEXT command had not specified a particular question, the COLLECT and KEYPUNCH programs would go to the next question in the questionnaire. The column on the right shows how this question is printed, complete with skip instructions, by the CONVERT program.

There are several points that you should keep in mind when using GOTO commands:

- GOTO commands may be used only with multiple choice, check-all-that-apply, and random number questions.
- The GOTO command must be written on the line immediately following the answer to be linked.
- Only one GOTO command may be linked to an answer.
- If an answer does not have a GOTO command linked to it, the skip will be determined by the NEXT command.

IF Command

As mentioned above, IF commands can be used to write more complex skip instructions, which are based on the responses to one or more questions. After the interviewer enters a response to a question, the COLLECT and KEYPUNCH programs will evaluate any IF commands that may have been used with that question to determine which question should be asked next.

Example

Source Program	Questionnaire Format
.QUESTION = Q1, TYPE = NUMBER	1. ...
...	...
...	...
...	...
.ANSWER = 5	_ _ _ _ _ _ _
.IF (Q1 < 100) Q20	
.IF (Q1 < 200) Q21	
.NEXT = Q22	IF (#1 < 100) GO TO QUESTION 20 IF (#1 < 200) GO TO QUESTION 21 SKIP TO QUESTION 22 =====

In this example, two IF commands have been used in question Q1. They will be evaluated after the interviewer has entered an answer to Q1. If the interviewer entered a number less than 100, the test expression for this first IF command will be true and the COLLECT and KEYPUNCH programs will skip to question Q20. If the interviewer entered a number that was greater than or equal to 100, the test expression will be false and the next IF command will be evaluated. If the interviewer had entered a number that was less than 200 (but still greater or equal to 100), the programs will skip to question Q21. Finally, if the interviewer entered a number that was greater than or equal to 200, the COLLECT and KEYPUNCH programs will skip to the question specified by the NEXT command. In this example, it will skip to question Q22. (If no question had been specified at the NEXT command, they would skip to the next question in the list.)

IF Rules

There are several important points that you must keep in mind when writing IF commands:

- IF commands must be used within a question. That is, they must be written after the QUESTION command and before the NEXT command.
- More than one IF command may be used in a question. The COLLECT and KEYPUNCH programs will skip to the question specified by the first test expression that evaluates to a true value.
- IF commands may be mixed with GOTO, SKIP, and SKIPIF commands within the same question. The COLLECT and KEYPUNCH programs will evaluate the commands in the order that they were used and skip to the question specified by the first command that has a true test expression.¹

¹This processing order differs from the QPL Version 2.0 software. In that version, all of the GOTO commands were checked before any IF commands, regardless of the order in which the commands were used.

- IF commands should refer only to questions that the interviewer has answered. You cannot be assured of a particular value for a question if it is not on the path that the interviewer is on. Questions on other paths, or questions that are farther down the path, may not contain the default answers you originally specified because the interviewer may have gone down that path entering data, then backed up and proceeded down a new path.
- You should not use the DATE or TIME keywords in IF commands. Although they would appear to work correctly during the interview, they will not work as expected when you reload a completed interview with the KEYPUNCH program because the current date and time will be different from the original interview. If you need to specify a skip based on the current time or date, you should instead refer to the current value of a TIME or XDATE question. Once the interviewer passes through these questions, the time or date is recorded and may not be changed, even if the record is later reloaded with the KEYPUNCH program.
- You must write test expressions using the values of questions as they are stored internally. For example, dates are stored as single integer in either YYMMDD or YYYYMMDD format, and times are stored as the number of seconds that have elapsed since midnight.²

You should always try to simplify the organization of your questionnaire before using additional IF commands. Although the COLLECT and KEYPUNCH programs will evaluate as many IF commands as you care to write, understanding the results of the analysis of complex questionnaires can become very difficult.

SKIPIF Command

The SKIPIF command works like the SKIP command, but allows you to add a test expression to determine whether it should compute the number of a question and skip to it.

Example

Source Program	Questionnaire Format
<pre>.QUESTION = R1, TYPE = RANDOM Pick a random starting question. .ANSWER First Second Third Fourth .SKIPIF (R1 > 0) (!Q1 + R1 - 1) .NEXT = FINISH</pre>	<pre>1. Pick a random starting question. __ 1. First __ 2. Second __ 3. Third __ 4. Fourth SKIP IF (#1 > 0) (2 + #1 - 1) SKIP TO QUESTION 6 =====</pre>

²See chapter 3.

In this example, question R1 from the earlier example of the SKIP command has been rewritten using the SKIPIF command. Instead of using an IF command to make sure that the random number question has been set (that is, the interviewer has passed through the question), the value of the R1 is checked before the question number is calculated. If R1 is zero, the test expression will be false, and the COLLECT and KEYPUNCH programs will skip to the question specified at the NEXT command.

Equation Processor

Many QPL commands use expressions to perform a variety of tasks. For example, they can be used to compute a value for a question, determine when a skip should be performed, and determine if an interview record should be saved. QPL expressions are similar to other programming languages or spreadsheet software. There are only a few rules that govern how expressions must be written that may be different from other systems:

- An expression must be enclosed in parentheses. This tells the COMPILE program where the expression begins and ends on a command line, and separates the expression from other expressions or commands that may be on the same line.
- An expression must be on a single line. The COMPILE program will give you an error message if the expression is split across two or more lines.
- Run-time math errors are defined to have a value of zero. For example, if a divide-by-zero error occurs in an expression, the equation processor will give that part of the expression where the error occurred a value of zero and then continue processing the expression.³

Operators

QPL expressions are evaluated from left to right, according to the following hierarchy of operations.

³This is done because it is not very useful to give the interviewer a divide-by-zero error message during the middle of an interview. You can use IF commands, for example, to check the value of variables that could cause an undefined expression. Syntax errors in expressions will be caught by the COMPILE program.

QPL Hierarchy of Arithmetic Operators

1. **NOT, functions**
2. **^ or **** (exponentiation)
3. ***, /, @**
4. **+, -**
5. **<, >, <= or =<, >= or =>**
6. **=, <> or ><**
7. **AND**
8. **OR**

Note: The @ operator is unique to the QPL equation processor. It is used to determine which answers in a check-all-that-apply question have been checked by the interviewer. See examples in chapters 3 and 5.

As this table shows, for example, multiplication and division are performed before addition or subtraction, and both of these operations are done before any tests of equality are made. This hierarchy can be overridden by using parentheses to group operations. There is no limit on how many parentheses may be used. The following example shows how parentheses may be used to control the order of evaluation.

Example

```
(3 + 4 * 5) evaluates to 23  
  
((3 + 4) * 5) evaluates to 35
```

In the first example, 4 is multiplied by 5 first because multiplication is a higher order operation than addition. Then 3 is added to the result, totaling to 23. In the second example, parentheses are used to force the addition to be done first. Here, 3 and 4 are added together, then the sum is multiplied by 5, resulting in an answer of 35. You may always use parentheses to group operations if you are unsure about how the QPL equation processor will order operations.

Variables

Any type of QPL question that records an answer as a number may be used as a variable in an expression. These include MULT, CHECK, RANDOM, NUMBER, DATE, LDATE, XDATE, and TIME questions. Simply use the name of a question as a variable when writing an expression. The COLLECT and KEYPUNCH programs will substitute the current value of the question for the name when they evaluate expressions.

You must remember, however, how the answers to questions are recorded in the data file when testing or computing the values of questions. For example, multiple choice and random number question answers are coded using numbers beginning at one. Check-all-that-apply questions are stored as a binary number.

LDATE questions are stored in YYYYMMDD format, which makes it easy to compare ranges of dates. Although DATE and XDATE answers are stored in YYMMDD format, they are promoted to YYYYMMDD format before they are used in an expression. And the TIME question is stored as a the number of seconds that have elapsed since midnight.

Example

Source Program	Questionnaire Format
.QUESTION = Q1, TYPE = DATE	1. ...
...	
.ANSWER	19 _ _ - _ _ - _ _
.IF (Q1 = 19910704) Q7	Year Month Day
.NEXT	IF (#1 = 19910704) GO TO #7

In this example, an expression is used with an IF command to check the date that the interviewer entered. Since the date is used internally as one integer in YYYYMMDD format, the test expression must use the same format. If the interviewer entered the date, July 4, 1991, this expression will be true, and the COLLECT and KEYPUNCH programs will skip to question Q7.

Keywords

In addition to numbers and variables, there are several QPL keywords that may be used in expressions. Keywords are used to get the interviewer's password number, the current time or date, a "Don't know" or missing value, true and false values, or the number of the last question the interviewer answered before pressing the Escape key.

QPL Expression Keywords

Keyword	Return Value
COUNT	A unique case identification number if interviewer is working on a new case or zero if the case was reloaded from the data file (range 1 to 32,000)
DATE	The current date (in YYYYMMDD format)
ESCAPE	The number of the question that was displayed before the Esc key was pressed
FALSE	A number whose value is false (zero)
MISSING or ?	A number that has a missing value
NEW	True if the interviewer is working on a new case or false if the case was reloaded from the data file
OPTION	Number entered at the DOS command line using the /O# option when the program was started.
PASSWORD	The interviewer's password number
RECORD	The current record number
TIME	The current time (as seconds since midnight)
TRUE	A number whose value is true (any non-zero number).

Notes: The TIME and DATE keywords should be used only with SHOWIF, COMPUTE, or COM-

PUTEIF commands. The RECORD keyword is available only when writing expressions interactively using various menu functions in the KEYPUNCH and CONVERT programs.

Keywords are used in an expression just like the names of questions. The COLLECT and KEYPUNCH programs will replace a keyword with a value when they evaluate an expression.

Example

Source Program	Questionnaire Format
.QUESTION = Q1, TYPE = NUMBER	1. ...
...	
.ANSWER = 5	_ _ _ _ _ _ _
.IF (Q1 = MISSING) Q8	
.IF (PASSWORD > 100) Q23	IF (#1 = MISSING) GO TO #8
.NEXT	IF (PASSWORD > 100) GO TO #23

In this example, the first IF command uses the MISSING keyword to check whether the interviewer has entered a “Don’t know” response by pressing the question mark key. The equation processor will treat a “Don’t know” response as zero unless you specifically compare it to the MISSING keyword using the = or <> operators. If the interviewer typed a question mark, this expression would be true, and the COLLECT and KEYPUNCH programs will skip to question Q8.

The second IF command checks the interviewer’s password number. You may require the interviewer to start the COLLECT or KEYPUNCH programs with a password number by using the PASSWORD command.⁴ If the interviewer’s password number is greater than 100, the programs will skip to question Q23; otherwise they will skip to the next question in the list.

COUNT and NEW Keywords

The COUNT keyword will return a unique case identification number for a new case. By default, it will begin with a value of one. Every time a questionnaire is started, a new COUNT value is generated, even if a questionnaire is not saved. Gaps in the sequence of numbers that are stored in a NUMBER question indicate that an interviewer started a new questionnaire but did not save the answers.

If an interviewer recalls an existing questionnaire record the COUNT keyword will return the value zero. You can prevent the previous value from being overwritten by using the NEW keyword as shown in the example below.

⁴See chapters 9 and 10.

Example

Source Program	Questionnaire Format
.QUESTION = ID, TYPE = NUMBER	1. ...
...	
.ANSWER = 4	_ _ _ _
.COMPUTEIF (NEW) COUNT	
.NEXT	COMPUTE IF (NEW) COUNT

In this example, the COMPUTEIF command is used to check whether this is the first time this questionnaire record has been used (that it did not already exist in the data file). If it is a new record, COUNT will return a unique case identification number which will be used as a response to this question. If it is not a new record (it was reloaded from the data file), the NEW keyword will be false and the COUNT value will not be used as a response—leaving the previous response intact. In other words, the value of question ID is set the very first time the interview was started. It cannot be changed later if the interview is reloaded into the COLLECT or KEYPUNCH programs.

The COLLECT and KEYPUNCH programs automatically create a file to store the last value of the COUNT keyword. This file will have the same name and be located on the same disk drive and directory as your questionnaire but will have a “CNT” extension. This is an ASCII text file that only contains the last COUNT number. You may change the starting value from by editing it with the DOS EDIT or Windows Notepad programs. The starting number must be located at the very beginning of the file.

If your interviewers will be making many calls (and attempts), you should add a check to make sure that the COUNT value does not exceed the storage capacity of the 4-digit number field.

Example

Source Program	Questionnaire Format
.QUESTION = ID, TYPE = NUMBER	1. ...
...	
.ANSWER = 4	_ _ _ _
.COMPUTEIF (NEW) COUNT	
.IF (COUNT > 9500) WARN	COMPUTE IF (NEW) COUNT
.NEXT = Q1	IF (COUNT > 9500) #2
	Skip to question #3
	=====

```
.QUESTION = WARN, TYPE = VOID      2. Warning! Please inform the
Warning! Please inform the         project manager that the ID
project manager that the ID        will soon exceed maximum the
will soon exceed the maximum       value of 9,999.
value of 9,999.
.ANSWER                             Press ENTER to exit...
Press ENTER to exit...
.NEXT
```

In this example, an IF command has been added to question ID. If the value of COUNT exceeds 9,500 the COLLECT and KEYPUNCH programs will skip to question WARN, which displays a message to the interviewer warning him or her of the potential problem. If COUNT does not exceed 9,500, the warning message will not be displayed.

OPTION Keyword

The OPTION keyword is used to customize a questionnaire based on a command line option that was used to start the COLLECT or KEYPUNCH programs. The value of the OPTION keyword is set by using the /O command line option when starting the COLLECT or KEYPUNCH programs. You can use this value in your questionnaire to compute a value to a question or as a test value in a skip instruction.

Example

```
DOS Command Line
C:\> COLLECT JOB1.QPL /O2

Source Program                Questionnaire Format

.QUESTION = Q1, TYPE = MULT   1. ...
...
.ANSWER                       (CHECK ONLY ONE ANSWER)
Version one
.GOTO = V1                    |__| 1. Version one (GO TO QUESTION 12)
Version two                   |__| 2. Version two (GO TO QUESTION 13)
.GOTO = V2                    |__| 3. Version three (GO TO QUESTION
14)
Version three
.GOTO = V3
.COMPUTE (OPTION)
.NEXT
```

In this example, the interviewer is automatically sent down the second path of this questionnaire. Since the value of OPTION is set to 2 when the COLLECT program was started (using the /O2 command line option), the COMPUTE command will set the answer to question Q1 to 2, without displaying this question to the interviewer. Next, since the second answer, "Version two," has a GOTO command attached to it, the COLLECT program will automatically jump to question V1.

Functions

The QPL equation processor also has a number of date and math functions that you can use to create more complex expressions. Using the date functions, for example, you can compute the number of days between dates, determine what day of the week a date falls on, or whether a date falls on a weekend or U.S. federal holiday. The math functions provide a way to round numbers, compute logs and antilogs, and find square roots.

You may put a constant, a variable name, an expression, or another function within the parentheses following the function name. The COLLECT and KEYPUNCH programs will first evaluate the expression inside the function's parentheses, and then use the result to calculate the function. All of the functions will return zero if they are given invalid arguments. You should check the value of any question that could cause an error before using it as an argument for one of these functions.

Date Functions

TOJUL()	Return Julian date from Gregorian date
FROMJUL()	Return Gregorian date from Julian date
JDAYOFWEEK()	Return a code that indicates the day of week from a Julian date
JFHOLIDAY()	Return a code that indicates a weekend or U.S. federal holiday from a Julian date
JYEAR()	Return year from Julian date
JMONTH()	Return month from Julian date
JDAY()	Return day from Julian date
GYEAR()	Return year from Gregorian date
GMONTH()	Return month from Gregorian date
GDAY()	Return day from Gregorian date

Notes: Arguments used with functions that begin with the letter J, and the FROMJUL() function, must be Julian dates. Arguments used with functions that begin with the letter G, and the TOJUL() function, must be Gregorian dates written in YYYYMMDD format. Answers to DATE and XDATE questions are automatically promoted to YYYYMMDD format when used in expressions.

The QPL equation processor uses Julian dates to perform date arithmetic. Julian dates differ from normal, or Gregorian, dates because they use one integer to represent a particular day instead of three. After you have converted dates to their Julian equivalents, you can, for example, find the number of days between two dates by subtracting one from another.

While many software packages use something similar to Julian dates for this purpose (such as the Lotus @DATE function), the QPL equation processor uses the same Julian dates that are used in astronomy, with the one variation that the Julian days used here begin at midnight instead of noon.⁵ The Julian date func-

⁵The QPL Julian date functions are based on the C functions by Flanery, Teukolsky, and Vetterling, *Numerical Recipes in C*, pp. 10-13, and Clark, "A Date Object in C++," *The C Users Journal*, pp. 57-69, June 1990.

tions used here consider leap years and the 10-day gap that occurred in October 1582 when the Julian calendar was converted to the Gregorian calendar. They can accommodate dates ranging from January 1, 0001, through December 31, 9999.

Math Functions

ABS()	Return absolute value of a number
SGN()	Return sign of a number (1 if positive or zero, 0 if negative)
ROUND()	Return integer that has nearest value to a real number
FLOOR()	Return largest integer that is less than or equal to a real number
CEIL()	Return smallest integer that is greater than or equal to a real number
MOD()	Return the decimal portion of a real number
LOG()	Return the natural log of a number
EXP()	Return the exponential of a number (the inverse of the natural log)
LOG10()	Return the base-10 log of a number
SQR()	Return the square root of a number

**TOJUL() and FROMJUL()
Functions**

The TOJUL() function converts a Gregorian date, in YYYYMMDD format, to an equivalent Julian date. For example, July 4, 1991, has a Julian date of 2,448,442. Once you have put a date into Julian format, you can add or subtract days from it to find new dates, or subtract the date from another Julian date to find the number of days between two dates.

The FROMJUL() function converts a Julian date back to a Gregorian date in YYYYMMDD format. This function allows you to perform arithmetic on Julian dates, and then convert the result to Gregorian again.

Example

Expression	Result
TOJUL(19910704)	2,448,442 (Julian date)
TOJUL(19910704)-TOJUL(19910604)	30 days
FROMJUL(TOJUL(19910604)+30)	19910704 (July 4, 1991)
FROMJUL(2448442)	19910704

JDAYOFWEEK() Function

The JDAYOFWEEK() function finds the day of the week that a Julian date falls on. It returns a code number that represents the name of the day. These codes are listed in the following table.

JDAYOFWEEK() Return Codes

- 0 Invalid date
- 1 Sunday
- 2 Monday
- 3 Tuesday
- 4 Wednesday
- 5 Thursday
- 6 Friday
- 7 Saturday

The following example shows how this function could be used to compute an answer to a multiple choice question that is later used to customize a question for a particular respondent.

Example

Source Program	Questionnaire Format
.QUESTION = Q1, TYPE = DATE When were you born? .ANSWER .NEXT	1. When were you born? 19 _ _ - _ _ - _ _ Year Month Day
.QUESTION = Q2, TYPE = MULT Compute day of week on which respondent was born. .ANSWER Sunday Monday Tuesday Wednesday Thursday Friday Saturday .COMPUTE (JDAYOFWEEK(TOJUL(Q1))) .NEXT	2. Compute day of week on which which respondent was born. (CHECK ONLY ONE ANSWER) _ 1. Sunday _ 2. Monday _ 3. Tuesday _ 4. Wednesday _ 5. Thursday _ 6. Friday _ 7. Saturday COMPUTE (JDAYOFWEEK(TOJUL(#1)))
.QUESTION = Q3, TYPE = VOID You were born on a [Q2]. .ANSWER Press ENTER to continue... .NEXT	3. You were born on a #2. Press ENTER to continue...

JFHOLIDAY() Function

The JFHOLIDAY() function returns a code that indicates whether a Julian date falls on a weekday, weekend, or U.S. federal holiday (the observed date during the week, according to the current law). If the date falls on holiday, the code also indicates which holiday it is. These codes are listed in the following table.

JFHOLIDAY() Return Codes

- 0** Weekday (non-holiday)
- 1** Weekend
- 2** New Year's Eve
- 3** Martin Luther King Day
- 4** President's Day
- 5** Memorial Day
- 6** Independence Day
- 7** Labor Day
- 8** Columbus Day
- 9** Veterans Day
- 10** Thanksgiving
- 11** Christmas

Like the JDAYOFWEEK() command, you could use this function to compute an answer to a multiple choice question.

Example (continued)

Source Program	Questionnaire Format
<pre>.QUESTION = Q4, TYPE = MULT Determine if birth occurred on a holiday. .ANSWER a weekday a weekend New Year's Eve Martin Luther King Day President's Day Memorial Day Independence Day Labor Day Columbus Day Veterans Day Thanksgiving Christmas .COMPUTE (JFHOLIDAY(TOJUL(Q1))+1) .NEXT .QUESTION = Q5, TYPE = VOID You were born on [Q4]. .ANSWER = Q3 .NEXT</pre>	<pre>4. Determine if birth occurred on a holiday. (CHECK ONLY ONE ANSWER) __ 1. a weekday __ 2. a weekend __ 3. New Year's Eve __ 4. Martin Luther King Day __ 5. President's Day __ 6. Memorial Day __ 7. Independence Day __ 8. Labor Day __ 9. Columbus Day __ 10. Veterans Day __ 11. Christmas COMPUTE(JFHOLIDAY(TOJUL(#1)) + 1) 5. You were born on #4. Press ENTER to continue...</pre>

**JYEAR(), JMONTH(), and
JDAY() Functions**

The JYEAR(), JMONTH(), and JDAY() functions are used to extract the year, month, or day information from a Julian date.

Example

Expression	Result
JYEAR(2448442)	1991
JMONTH(2448442)	7 (July)
JDAY(2448442)	4 (the fourth)

Note: The number 2,448,442 is the Julian date for July 4, 1991.

GYEAR(), GMONTH(), and GDAY() Functions

The GYEAR(), GMONTH(), and GDAY() functions are used to extract the year, month, or day information from a Gregorian date that is written using YYYYMMDD format.

Example

Expression	Result
GYEAR(19910704)	1991
GMONTH(19910704)	7 (July)
GDAY(19910704)	4 (the fourth)

ABS() and SGN Functions

The ABS() and SGN() functions are useful when using negative numbers. The ABS() function returns the absolute value of a real number. The SGN() function will return a true value (1) if a number is zero, positive, or not known (missing). It will return false value (0) if it is negative.

Example

Expression	Result
ABS(-123.45)	123.45
ABS(123.45)	123.45
SGN(-123.45)	0
SGN(0)	1
SGN(123.45)	1

ROUND() Function

The round function returns the integer that is closest, or equal to, a real number. The fractional portion of the real number will be rounded up if it is 0.5 or larger, and it will be rounded down if it is less than 0.5.

Example

Expression	Result
ROUND(10.9)	11
ROUND(10.5)	11
ROUND(10.4)	10
ROUND(-10.9)	-11
ROUND(-10.5)	-11
ROUND(-10.4)	-10

FLOOR() and CEIL() Functions

The FLOOR() and CEIL() functions provide an alternative way of rounding real numbers. The FLOOR() function will return the “floor” value of a real number: the largest integer that is less than or equal to a real number. Conversely, the CEIL() function will return the “ceiling” value: the smallest integer that is greater than or equal to a real number.

Example

Expression	Result
FLOOR(10.9)	10
FLOOR(10.5)	10
FLOOR(10.4)	10
FLOOR(-10.9)	-11
FLOOR(-10.5)	-11
FLOOR(-10.4)	-11
CEIL(10.9)	11
CEIL(10.5)	11
CEIL(10.4)	11
CEIL(-10.9)	-10
CEIL(-10.5)	-10
CEIL(-10.4)	-10

MOD() Function

The MOD() function extracts the fractional portion of a real number. The sign of number is maintained.

Example

Expression	Result
MOD(3.14159)	0.14159
MOD(-3.14159)	-0.14159

LOG() and EXP() Functions

The LOG() and EXP() functions allow you to use natural logarithms in QPL expressions. The LOG() function returns the natural log of a real number, and the EXP() function returns the exponential, or inverse of the natural log.

Example

Expression	Result
LOG(100)	4.60517
EXP(4.60517)	100

LOG10() Function

The LOG10() function returns the base-10 log of a real number. The antilog may be computed using the power operator (^ or **).

Example

Expression	Result
LOG10(100)	2
10^2	100
10**2	100

SQR() Function

The SQR() function returns the square root of a positive number. It will return zero if the number is less than or equal to zero.

Example

Expression	Result
SQR(100)	10
SQR(-100)	0

Exiting the Questionnaire

Normally, the only way to quit the COLLECT program, or return to the command menu in the KEYPUNCH program, is to answer all the appropriate questions in the questionnaire. This feature is important because it preserves the skip logic of the questionnaire.

ESCAPE Command

Sometimes, however, a questionnaire cannot be completed, and it is necessary for the interviewer to quickly restart the program. This can be accomplished by programming the Escape key to arbitrarily jump to a particular question when it is pressed by the interviewer. It should be programmed to jump to one of the last questions that asks the interviewer whether he or she wants to save or erase the answers from this respondent. Programming the Escape key to jump to the end of the questionnaire is especially useful when using the KEYPUNCH program. This provides a quick way of exiting the questionnaire when only a few items need to be edited.

Example

```
.ESCAPE = Q42
```

In this example, the Escape key is set to jump to question Q42 when it is pressed. If the Escape key is not set to a particular question, it will jump to the last question in the questionnaire.

Pressing the Escape key does not change how the COLLECT and KEYPUNCH programs evaluate skip instructions. The answers that the interviewer entered, and the default answers to questions that were not answered, will still be used to

find the unique pathway through the questionnaire.⁶ The answers the interviewer entered and default answers will be stored in the data file, but the answers that were not on the path will be set to blank spaces (missing values).

Saving the Responses

Knowing when to save the responses to an interview, and when to discard them, can be sometimes be a difficult programming problem. If you do not do any special programming, the COLLECT program will save a record every time the interviewer starts the program. The KEYPUNCH program will save records only when the interviewer tells it to. As a rule, you should always provide the interviewer with some way to erase an interview, while it is still in the memory of the computer, or at least to mark the interview record as being incomplete or invalid.

Controlling when records are saved or dropped involves both writing a series of questions directed to the interviewer and using a special QPL command word that determines when to save a record. The questions simply ask the interviewer if he or she wishes to save or erase the responses to the interview that is currently in the memory of the computer. This does not, however, allow the interviewer to erase a record that has been saved in the data file. Removing records from the data file can be accomplished using menu functions in the KEYPUNCH or CONVERT programs.⁷

SAVEIF Command

After the interviewer has passed through the very last question, you can program the COLLECT and KEYPUNCH programs to read the response to the “save or erase” question and then act accordingly.⁸ This is done using the SAVEIF command. The SAVEIF command is similar to the IF command. Here, you may write an expression that will determine if a record will be saved or dropped. If the expression is true, it will add the record to the data file. If it is false, the record will be erased from the computer’s memory.⁹

The SAVEIF command may only be used once in your questionnaire program. It may be typed on any command line, but it is usually written near the beginning of the program. The following example shows how a series of ending questions would usually be written to give the interviewer control over when a record is saved.

⁶The ESCAPE keyword can be used in expressions to identify the question that was displayed when the Escape key was pressed. This information is needed when using control files to restart incomplete interviews at the point where they were interrupted. See chapter 6.

⁷See *QPL Data Editing Manual—Version 4.0*, and chapter 11.

⁸The KEYPUNCH program will display an error message if the interviewer tries to save a record that does not meet the SAVEIF specifications. The interviewer should erase the record in memory when this occurs.

⁹In QPL Version 2.0, the SAVEIF command also determined when control file records would be updated. In this version, the SAVEIF command does not affect control file records. A new command, SAVECONTROLIF, is used to determine when control file records should be updated. See chapter 6.

Example

Source Program	Questionnaire Format
.ESCAPE = FINISH .SAVEIF (FINISH = 2)	
.QUESTION = FINISH, TYPE = MULT You have finished the questionnaire. You may now review, save or erase your answers. .ANSWER Review .GOTO = Q1 Save .GOTO = ENDTIME Erase .NEXT	47. You have finished the questionnaire. You may now review, save or erase your answers. __ 1. Review (GO TO QUESTION 1) __ 2. Save (GO TO QUESTION 49) __ 3. Erase
.QUESTION = MAKESURE, TYPE=MULT Are you sure you want to erase your answers? .ANSWER = 2 Yes No .GOTO = FINISH .NEXT	48. Are you sure you want to erase your answers? __ 1. Yes __ 2. No (GO TO QUESTION 47)
.QUESTION = ENDTIME .TYPE = TIME Time questionnaire ended. .ANSWER .NEXT	49. Time questionnaire ended. _ _ _ _ _ _ _ _ SAVE IF (#47 = 2)

In this example, the Escape key is programmed to jump to the question called “FINISH,” from anywhere in the questionnaire. This is a multiple choice question that asks the interviewer whether he or she wants to save or erase the answers to the questionnaire. If “Erase” is chosen, a second question, called MAKESURE, is asked to verify that the interviewer does intend to erase the responses. Note that the default answer to MAKESURE is set to “No,” so that the interviewer cannot accidentally press the Enter key twice and fall through the question without answering it.

If the interviewer selects “Save,” then the GOTO command will make the COLLECT and KEYPUNCH programs skip directly to the last question, called “ENDTIME.” Since this is a time question, the computer will automatically fill in the time that the questionnaire was completed.

If the interviewer selects “Review,” the COLLECT and KEYPUNCH programs will skip back to the beginning of the questionnaire. From there, the interviewer may press the Escape key to return to the FINISH question (or proceed question-by-

question through the completed interview again). If the interviewer presses the Escape key, the FINISH question will be displayed. It is important to note, however, that pressing the Escape key at this point will have no effect on the questions that have been answered. The COLLECT and KEYPUNCH programs will determine the skip path through the questionnaire according to all the questions that have been answered, even though the Escape key was used to jump directly to the end of the questionnaire.

After the ENDTIME question is answered, the COLLECT and KEYPUNCH programs will evaluate the expression in the SAVEIF command to determine whether to save the record. In this case, the SAVEIF command has been programmed to save the record if the interviewer selected “Save” (*i.e.*, the second response) to FINISH.

If you do not use the SAVEIF command, the COLLECT program would save every record. But, if you have used a question such as FINISH, you still can screen unwanted records. If you are analyzing the data with SPSS-x, for example, you could use its SELECT IF statement to use only records in which the interviewer answered “Yes” to question FINISH.¹⁰

Example - SPSS-x

```
SELECT IF (FINISH EQ 2) /* Keep only "saved" records */  
  
/* Statistical procedures */
```

In this example, only records that have been marked to be saved by the interviewer (*i.e.*, coded 2 by the COLLECT and KEYPUNCH programs) will be used in an analysis. This is essentially equivalent to using the SAVEIF command in your questionnaire program itself, but it gives you the flexibility of deciding later whether you want to erase a record. If you erase it before it is saved in the data file, using the SAVEIF command, it is gone forever. Which method you choose depends on your application.

¹⁰The KEYPUNCH program allows the interviewer to choose when to save records from the command menu. It will not, however, allow a record to be saved if a SAVEIF command has been used and evaluates to a false value.

Data File Format

Interviews are always recorded on your disk in what is generally referred to as a “fixed format” data file. This is a generic type of data file that can be used by many different analytical software packages, on both micro and main frame computers. The QPL CONVERT program automatically generates SPSS, SAS, and BASIC programs that read this type of file. (See chapter 12.) Other statistical software packages also can be programmed to use this data file.

If you are using Lotus or dBase to analyze the data, you can use the CONVERT program to translate a QPL fixed format data file directly into their data file formats. (See chapter 12.)

Fixed Format Files

Fixed format data files are fundamentally different from the data files that Lotus and dBase use. Lotus and dBase data files contain information on how the data are stored. When they read one of their files, they automatically know what columns, rows, or variable names to assign to each piece of data. Fixed format files contain no information about how data are stored. Any program that accesses data from a fixed format file must know where to look for individual data elements.

The data in fixed format data files are organized into “records,” “cards,” and “columns.” One record corresponds to all the data from one interview. The concept of a data card is a hold-over from the days of 80-column IBM paper punch cards. Each record may have one or more cards to store all the data for one record. Data from individual questions, such as a name or address, are defined to be written in a range of columns, character-by-character, on a card. The data are always written using ASCII codes, and each line is terminated by line feed (code 10) and carriage return (code 13) codes.

Example

Question	Column Locations		Fixed Format Card
	Start	End	
1. Date: February 10, 1987	1	6	87021028800ABC301
2. Time: 8:00 a.m.	7	11	-----
3. String: ABC	12	14	1 2 3 4 5
4. Number: 30	15	16	
5. Multiple Choice: Yes (First answer in list)		17	

This example shows how five answers would be written to a fixed format file by the COLLECT or KEYPUNCH programs. The date is stored in the first six columns

of the card (in YYMMDD format), the time is stored in the next five columns (as the number of seconds since midnight), a three-letter string is stored in the next three columns, and so on.

Generally, you do not have to be concerned with where the data from your interview are stored in a data file. The QPL COMPILE program automatically determines where data from each question will be written on imaginary IBM cards when the COLLECT or KEYPUNCH program saves the answers to an interview. The CONVERT program automatically writes SPSS and SAS programs that define the organization of data file, and translates the data into another format.

CARD Command

The COMPILE program will keep writing data elements to an 80-column card until it runs out of room. Then it will begin using a new card, up to a maximum of 32,000 cards. There are times, however, when you will need to change the default card size. For example, if you have defined a field for a question that is wider than the default of 80 columns, you will need to increase the size of the data card to at least the size of your largest field. You can increase the card size up to 32,000 columns using the CARD command.¹ This command can be used anywhere in your questionnaire program, but it is generally used near the beginning.

Example

Source program	Questionnaire Format
.CARD = 132	
.QUESTION = Q1, TYPE = STRING	1. ...
...	...
...	...
...	1:1-100
.ANSWER = 100	_____
.NEXT	_____

In this example, Q1 is a 100-character string question. Thus, the card size needs to be increased from the default of 80 columns to at least 100 columns. Here, the card size has been set to 132 columns. The formatted version of this question is shown on the right, including the location of this data element in the data file. It shows that data from this question always will be written on the first data card, from column 1 to column 100.

¹SPSS/PC+ allows a maximum card size of 1,024 columns; SAS/PC and BASIC allow 32,000 columns.

FIX Command

A error in the CARD command in Version 2.0 caused answer fields to be wrapped to the next data card one column too soon. For example, if the default card setting of 80 columns was used, the COMPILE program would wrap data fields to the next card after the 79th column was used instead of the 80th column, unless a STRING question was set to exactly 80 columns.

Because the CARD command in Version 1.0 worked correctly, the record layouts of the same program compiled under each version could be different, and thus you could not use a data file that was created with Version 1.0 with Version 2.0.²

This error has been fixed in Version 3.0 and 4.0. If possible, each data card will be filled to the 80th column (assuming the default CARD size). Now the record layouts of data files created with Version 3.0 or 4.0 will match Version 1.0, but not Version 2.0. You can, however, tell the Version 4.0 COMPILE program to wrap fields at the same point as the Version 2.0 COMPILE program by using the FIX command.

Example

```
.FIX = OFF
```

As this example shows, setting the FIX to OFF tells the COMPILE program to use the same record layout as Version 2.0. The default setting is ON, which specifies a Version 1.0, 3.0, or 4.0 record layout.

NEWCARD Command

In addition to changing the card size, there also may be instances when you need to control when the COLLECT and KEYPUNCH programs will begin writing data to the next data card. You may need to do this if you are trying to match the record layout of your questionnaire program to the record layout of a data file that was produced from another system. You can force the COLLECT program to begin using a new card early, before all the room is used up on a card, by using the NEWCARD command.

Example

```
Source Program                PUBLISHED Questionnaire
. CARD = 132
. QUESTION = Q1, TYPE = NUMBER 1. ...
  ...                          ...
  ...                          ...
  ...                          ...
. ANSWER = 5                    |_|_|_|_|_|_|
. NEXT
```

²Generally, however, this problem could be overcome by setting the CARD command to 81 instead of 80.

```
.QUESTION = Q2, TYPE = STRING      2. ...  
.NEWCARD                            ...  
  ...                               ...  
  ...                               2:1-100  
  ...                               _____  
.ANSWER = 100                       _____  
.NEXT                               _____  
                                   _____
```

A NEWCARD command is used in question Q2 to make COLLECT and KEYPUNCH programs write the answer to this question at the beginning of a new data card, even though there is room available on the first card. This does not waste as much space as it would first appear, however, because QPL data cards are not padded to the end with blank spaces. Carriage return and line feed codes are used to end each card immediately after the last data element. Thus, the total record size of the two questions in the example would be 109 bytes.

Example

```
5 digit number  
1 line feed  
1 carriage return  
100 character string  
1 line feed  
+ 1 carriage return  
-----  
109 Total record length
```

Missing Values

Before the COLLECT and KEYPUNCH programs add a new record to a data file, they set all of the answers that were not answered, according to the skip instructions, to a missing value code. The entire field for an answer is set to the missing value code. The QPL missing value code is always a blank space (ASCII code 32).

An answer to a number or date question also may be set to blank spaces if the interviewer entered a “Don’t know” response, by pressing the question mark key.

The use of missing value codes greatly simplifies the analysis of the questionnaire data because SPSS and SAS automatically do not consider blank answers when calculating frequency or descriptive statistics.

Answer Field Sizes

The number of columns each question uses on a card in the data file is determined when you program each question. Number and string questions require that you explicitly state how many columns, or characters, be used to store a response. Other types of questions either use a fixed number of columns or automatically determine how many columns are needed. Generally, you should try and keep your data record as compact as possible. This provides several benefits:

- more records can be stored on a disk,
- the data file can be uploaded in less time, and
- the data file can be processed faster by analytical programs.

Multiple Choice and Random Number Questions

The COLLECT and KEYPUNCH programs automatically code the list of answers to multiple choice and random number questions beginning with the number one. The SPSS and SAS programs that are generated by the QPL CONVERT program automatically link these codes back to the original list of answers. When an answer is selected, by the interviewer or randomly by the computer, the COLLECT and KEYPUNCH program will write the number of the answer in the data field.

The COMPILE program will reserve one column to store the answer if the list contains fewer than 10 responses, since the answer number can only be one digit ranging from 1 to 9. Two digits will be reserved if the list has 10 or more responses. In this case, an answer that is less than 10 will be right-justified in the field. A blank space will be used to pad the left side of the field, if necessary. The whole field will be set to blank spaces if the question was skipped.

Check-All-That-Apply Questions

Check-all-that-apply question fields use one digit for each answer in the list to indicate whether a response has been checked. If one has been checked, its digit will be set to 1. Otherwise it will be zero. The first answer corresponds to the left-most digit in the field, and the last answer corresponds to the right-most digit. The entire field will be set to blanks if the question was skipped.

Example

Display on COLLECT Screen	Fixed Format File
Red	0110000
√ Orange	
√ Yellow	
Green	
Blue	
Indigo	
Violet	

This field is treated as a binary number when you refer to it in a QPL expression. The order of the digits, however, will be reversed when the COLLECT or KEYPUNCH programs find the current value of the field. The left-most digit, for example, will become the least significant bit, and the right-most digit will become the most significant bit.

Example

Field Contents	Binary Representation
0110000	0000110
- - - -	- - - -
1234567 (position in field)	6543210 (bit significance)
ROYGBIV (color)	VIBGYOR (color)

In this example, the value of the field when used in an expression will be 6, the decimal equivalent of the binary number 110. Normally, you do not have to be concerned with binary arithmetic because the QPL @ operator makes it relatively easy to determine which answers have been checked.

Example

```
.IF (Q1 @ 2 AND Q1 @ 3) Q9
```

This expression uses the @ operator to determine whether Orange and Yellow (the second and third responses) have been checked in the previous example. You can use this operator with the AND, OR, and NOT logical operators to test for any combination of check marks.

You can, however, use your knowledge of binary arithmetic to shorten some of your expressions.

Example

```
.IF (NOT(Q1@1 OR Q1@2 OR Q1@3 OR Q1@4 OR Q1@5 OR Q1@6 OR Q1@7)) Q9  
.IF (Q1 = 0) Q9
```

Both of these expressions test whether all of the answers were not checked. Since, in this case, all of the digits will be set to zero, the field will have a value of zero when used in an expression. Similarly, you could test for all of the answers being checked by comparing it to 127, the decimal equivalent of the binary number 111111.

You can compute the value of any combination of checks using the formula, “ $2^{n-1} + 2^{m-1} + \dots$,” where n and m are the numbers of the answers that are checked. For example, $2^{2-1} + 2^{3-1}$ equals 6, or the value of the field in the previous example when only Orange and Yellow are checked.

You do not need to be concerned with binary arithmetic when analyzing the results of your survey. Each digit in the field will be defined as a separate variable when you use the CONVERT program to create SPSS or SAS programs that read the data file. The SPSS and SAS programs will use each answer in the list as a variable name (instead of the question text) and use the words “Checked” and “Not checked” for value labels (instead of the answer list). Each digit also will be defined separately when you translate the data file to a Lotus or dBase file.

Number Questions

Only the digits the interviewer types, and a minus sign if necessary, are stored in a number question field. Decimal points and any other formatting information are not included. This information, however, is used by the CONVERT program when it defines the data file in an SPSS or SAS program, and when it translates the data into other formats, such as Lotus or dBase. Numbers are always right-justified in the data field. The left side of the field will be padded with blank spaces, and the minus sign floats in front of the number. If the interviewer sets a number to “Don’t know” by pressing the question mark key, the field will be filled with blank spaces. It will also be blank if the question was skipped.

Example - Five-Digit Number With Two Decimal Places

Display on COLLECT Screen	Fixed Format File
.00	___00
.01	___01
1.20	__120
-1.20	_-120
12.34	_1234
300.00	30000
___._?	_____ (Don't know response)

Note: The underline indicates blank spaces.

Short Answer Questions

Answers to string questions are always saved in the data file with same number of characters that was defined in the field size. If it was defined without a format, the field in the data file will be exactly the same as it appears on the screen. Blank spaces will fill any empty parts of the string. If a format was used, only the data entered at the template characters will be saved in the data file. The entire field will be set to blank spaces if the question was skipped.

Example - Unformatted Five-Character String

Display on COLLECT Screen	Fixed Format File
A_____	A_____
ABEL_____	ABEL_____
_____A	_____A
BAKER	BAKER
?_____	_____ (Don't know response)

Note: The underline indicates blank spaces.

Example - Format String, "State: @@ Zip: #####"

Display on COLLECT Screen	Fixed Format File
State: VA Zip: 22305	VA22305
State: MN Zip: 55110	MN55110
State: _X Zip: _____6	_X_____6
State: ?_ Zip: _____	_____ (Don't know response)

Note: The underline indicates blank spaces.

Open-Ended Questions

Answers to open-ended questions are not stored in the main data file with the answers to other questions. Since the interviewer may enter a variable amount of text as a response, the answer to each open-ended question in each interview is stored in separate files. These files are always put on the same disk and in the same directory as the main data file, and they are always given an OTX extension.³

The QPL programs manage these files by keeping a copy of the file name in an open-ended question's answer field in the fixed format data file. The open-ended question field is always eight characters long. The file name is constructed when the interviewer first displays the question on the COLLECT or KEYPUNCH screen. Generally, the name is a combination of a unique case identification number (the first four characters) and the name of the question (the last four characters).⁴ Zeros will be used to pad the left side of the field if the identification number has fewer than four digits, and dashes will be used to pad the right side of the field if the question name has fewer than four characters.

³You should not edit OTX files with your word processor. They contain codes that are unique to QPL and will not be displayed properly. Use the CONVERT program to translate these files into a generic word processor file.

⁴You may use the default answer line to define a name that is not based on the case identification number and question name. This allows one OTX file to be used to hold the responses from two or more questions. See chapter 3.

Example

Open-Ended Question	Fixed Format File
Identification number: 12	0012Q4--
Question name: Q4	

The QPL programs will automatically add the disk drive letter, path name, and OTX extension to this name whenever they read or write to the file. For example, if the questionnaire program was being run on your C: drive and in the JOB1 directory, the full file name would be C:\JOB1\0012Q4--.OTX.

Date, Long Date, and Current Date Questions

DATE and XDATE questions are always stored in six-column fields, in YYMMDD format. The answers to these questions, however, are promoted to YYYYMMDD format when they are used as variables in expressions.⁵ LDATE questions are always stored eight-column fields in YYYYMMDD format. Its format is not changed when used as a variable in an expression.

Example

Question Type	Display on COLLECT Screen	Data File Format	Expression Value
DATE	<u>07-04-1991</u>	YYMMDD	YYYYMMDD
XDATE	(not displayed)	YYMMDD	YYYYMMDD
LDATE	<u>07-04-1991</u>	YYYYMMDD	YYYYMMDD

Notes: The underline indicates where digits may be typed. The display on the COLLECT screen may be different if your computer is configured for a country other than the United States. The data file formats and expression values, however, will not be changed.

The CONVERT program will automatically define dates in an SPSS or SAS program with a date format, and will translate the data file itself using the Lotus @DATE function or the dBase date type.⁶

⁵This implies, for example, that you must subtract 19,000,000 from a DATE or XDATE value in an expression to return the result back to YYMMDD format.

⁶The Lotus @DATE function does not allow years earlier than 1900 or later than 1999 to be entered. The CONVERT program will drop the first two digits of the year when it puts an LDATE field into a Lotus @DATE function.

Time Questions

Time questions are always stored in a five-digit field, as the number of seconds that have elapsed since midnight. It is always right-justified in the field, and zeros may be used to pad the left side of the field. The field will be set to blank spaces if the question was skipped.

Example

Time	Fixed Format File
12:01 a.m.	00060
8:00 a.m.	28800
8:30 a.m.	30600
12:00 noon	43200
3:30 p.m.	55800
6:15 p.m.	65700

Version Questions

The COLLECT and KEYPUNCH programs copy the answer line exactly as it is written to each record in the data file. Thus, the field size is determined by the number of characters in the answer line. The CONVERT program will define it as a string variable when it creates an SPSS or SAS program, as a label when it translates the data to a Lotus file, and as a character field when it translates the data to a dBase file.

Void Questions

Void questions do not record any information, so the COMPILE program does not reserve any space for them in the data file.

Scheduling Interviews

After you have written and tested your questionnaire and trained the interviewers to use the COLLECT program, your next biggest problem is keeping track of which respondents have been called. One solution to this problem is to keep records on paper, such as the respondent's name and phone number and a log of attempted calls. This requires, however, that the interviewer accurately copy the case number from the control sheet to the CATI program, and record the date and time that a call was completed on the paper log sheet.

Control Log Sheet

REMEDIAL EDUCATION IN THE
SUMMER YOUTH EMPLOYMENT AND TRAINING PROGRAM
(Job 205078)

*** Telephone Survey Respondent Control Sheet ***

|__|__|__| |_____|
Case ID SDA Name

\$|__|__|.|__|__|__| |\$|__|__|.|__|__|__|
1986 II-B Funding (millions) Unspent 1986 Funds (millions)

|_____|
Program Name

Phone Respondent's Name
|_____| |_____|
|_____| |_____|

Contact Date	Interviewer Initials	Result
__ /__ /87	__ __ __	_____
__ /__ /87	__ __ __	_____
__ /__ /87	__ __ __	_____
__ /__ /87	__ __ __	_____
__ /__ /87	__ __ __	_____
__ /__ /87	__ __ __	_____
__ /__ /87	__ __ __	_____
__ /__ /87	__ __ __	_____

Comments

(Write any additional comments on the back of this form.)

This is a copy of a call sheet that was used on a GAO audit of the Summer Youth Employment and Training Program. The information at the top of the form was filled out when the sample of respondents was identified from agency records. Funding information was also written in and then entered by the interviewer at the beginning of an interview. Each interviewer was given a stack of call sheets for each day's calls.

This method of managing the calls, although paper-intensive, does have some advantages. The supervisor and interviewers were easily trained on how to manage the process. Call sheets from completed calls were put in separate stacks from uncompleted calls, and the supervisor kept handing out call sheets until there were none left. And there was plenty of room on the form for the interviewer to describe any unusual circumstances about the call.

Automated Call Sheets

Another solution to this management problem is to use the QPL control file feature to create a database of respondents, which includes information on whether the call has been completed. When an interviewer starts a questionnaire using the COLLECT, NCOLLECT, or KEYPUNCH programs, the interviewer will be presented with a list of calls on the screen that have not been completed. When the interviewer selects one of the respondents, data about that respondent can be automatically copied from the control file to the questionnaire file. This information typically includes the respondent's name, title, phone number, and a case identification number.

This information also can be displayed on the screen for the interviewer. When the interviewer completes the call, the entry on that respondent can be updated to show that the call was completed. The next time the interviewer starts an interview, that respondent's name will not appear in the control list on the screen for the interviewer to choose from.

The control file feature is most conveniently used with the NCOLLECT program on a network. Here, all of the interviewers can share the same control list. NCOLLECT automatically prevents interviewers from attempting the same case and records the status of each case in one control list file.

Automated call sheets are a bit more difficult to use in a non-network environment (with the COLLECT program). Here, you must split your control list file into separate files for each interviewer in order to avoid duplicate calls. The

CONVERT program can be used to split a data file into two or more files.¹ These files can be later merged back together using CONVERT program or the DOS COPY command with the “+” option.

Creating and Using Control Files

The control file feature allows you to define complex relationships between the control file database and your questionnaire. You can define which respondents are displayed in the control list, and you can selectively copy information between the questionnaire and the control files. In short, you can customize your control file to fit almost any application.

Control files are just short QPL questionnaires. That is, you must first write a QPL program that defines the information that you want to include in your control data file, such as the respondent’s name, telephone number, case identification code, and a field that indicates whether the call was completed. When this is done, you can use the KEYPUNCH program to enter the information on each respondent. In order to test the control file program with your questionnaire, you will need to create a control data file. If you do not have the actual respondent information, create a file with dummy cases.

Control Commands Used in Control File Program

<hr/>	
<i>Required</i>	
HEADER	Define how control list will be displayed
SHOWIF	Define when a case will be displayed in control list
<i>Optional</i>	
IDENTIFY	Specify a question that holds the unique case identification number
RELOADIF	Define when an incomplete case should be reloaded

Next, you must add commands to your questionnaire program that tell the COLLECT, NCOLLECT, and KEYPUNCH programs to use this control file, and what pieces of data should be copied from the control file to the questionnaire file, such as the case identification number. You can also include commands that copy information, such as whether the call was completed, back to the control data file.

¹See chapter 12.

Control Commands Used in
Questionnaire Program

Required

CONTROL Set name of control file program

Optional

GET Copy answer from control file to questionnaire

PUT Copy answer from questionnaire to control file

COPY Copy answer from one question to another within a questionnaire
and allow editing

IDENTIFY Specify a question that holds the unique case identification number

INCREMENT Set control file question to increment after each attempted call

PAUSE Set question that will hold the number of the question that was
displayed before the Esc key was pressed

**SAVE-
CONTROLIF** Define when the control file record will be updated

The following example shows how a relatively simple control file program is constructed. Except for the use of the HEADER and SHOWIF commands, it is just a straightforward data-entry program.

Sample Control File Program

```
*****
** PROGRAM: C1.DOC                      DATE: 07/29/1991 **
** AUTHOR: KEVIN DOOLEY                  **
**                                       **
** TITLE: QPL Control File Demonstration I **
**                                       **
** PURPOSE: Demonstrate how to a CONTROL file to **
** restrict what cases may be entered.     **
**                                       **
** This program will create a data base of **
** respondents that a second program      **
** will use to control what cases the     **
** interviewer may enter. (In this example, **
** the data base will be made up of members of **
** the Washington Diplomats soccer team.) **
**                                       **
** This control file will be used with the **
** questionnaire program Q1.DOC. This program **
** demonstrates the minimal number of commands **
** required to set up a control file.     **
*****
** Global commands

.TITLE      = "QPL Control File Demonstration I"
.SUBTITLE   = "Version 4.0"
.ESCAPE     = FINISH
.UPPERCASE  = OFF
.SAVEIF     (FINISH = 1)
```

Chapter 6 Scheduling Interviews

```
*****
** Control file commands.

.HEADER      = "[NAME], [COLLEGE]"
.SHOWIF      (STATUS = 1)

*****
** Questions for control file data base.

.QUESTION = CASE, TYPE = NUMBER
.LOW = 1
Enter unique case identification number.
.ANSWER = 4
.NEXT

.QUESTION = NAME, TYPE = STRING
Enter respondent's name.
.ANSWER = 30
.NEXT

.QUESTION = COLLEGE, TYPE = STRING
Enter college or country.
.ANSWER = 40
.NEXT

.QUESTION = PLAYER, TYPE = NUMBER
Enter shirt number.
.ANSWER = 2
.NEXT

.QUESTION = PHONE, TYPE = STRING = "(###)###-####"
Enter respondent's phone number.
.ANSWER
.NEXT

.QUESTION = STATUS, TYPE = MULT
Status of call
(Select 'Outstanding' when creating data base.)
.ANSWER
Outstanding
Completed
.NEXT

*****
** Question used by SAVEIF to determine if data record
** should be saved.

.QUESTION = FINISH, TYPE = MULT
You may now save or erase this record.
.ANSWER
Save answers
Erase answers
.NEXT
```

Note: A copy of this program, C1.DOC, has been included on the QPL Master Program Disk.

SHOWIF Command

This example shows how a simple control file is constructed. Actually, only two new command words are needed: SHOWIF and HEADER. The SHOWIF command is used to define when a record will be displayed on the interviewer's screen

when he or she starts a new interview. In this case, it has been programmed to display a record from the control list database (called C1.DAT) if the answer to question STATUS is “Outstanding.”

More complex expressions may be used with the SHOWIF command. For example, you could also specify a specific time or day that a case may be eligible to be called.

The network NCOLLECT program further limits the cases that may be displayed to the interviewer. It will not display cases that are currently being used by another interviewer. This additional restriction prevents two or more interviewers from attempting the same case at the same time (even though the case is listed in the control file as “Outstanding”).

HEADER Command

The HEADER command defines how this record will be displayed in a list of records to the interviewer. The HEADER string is programmed just as if it was a question text line, except that the string must be delimited by double quotes. Square brackets with variable names define where the answers from questions will be shown, and other text is used to format the line. Here, the contents of two questions, NAME and COLLEGE, will be displayed to identify a record in the database. A comma and a blank space will be used to separate the two items on the line.

The rest of this program is just concerned with entering information about each respondent. The time, date, and version questions are routinely used with any QPL program as a defense against mistakes. This information allows you to trace how and when problems may have occurred during data-entry.

Merging the Control and Questionnaire Programs

Most of the work of using a control file is in the questionnaire program itself. Here, you must define:

- The name of the control file.
- What information should be copied from the control data file to the questionnaire.
- What information should be copied from the questionnaire to the control data file.
- When a completed questionnaire and changes to the control data file will be saved.

Questionnaire Program Using a
Control File

```
*****
** PROGRAM: Q1.DOC                                DATE: 07/29/1991 **
** AUTHOR: KEVIN DOOLEY                          **
**                                               **
** TITLE: QPL Control File Demonstration I       **
**                                               **
** PURPOSE: Demonstrate how to a CONTROL file to **
**           determine what cases may be entered. **
**                                               **
**           This program uses a data base of respondent **
**           names and phone numbers to control who **
**           the interviewer may call that were created **
**           with control file program C1.DOC.     **
**                                               **
*****
** Global commands.

.TITLE           = "QPL Control File Demonstration I"
.SUBTITLE        = "Version 4.0"
.ESCAPE          = FINISH
.HEADER          = "[NAME], [COLLEGE]"
.UPPERCASE       = OFF
.SAVEIF          (FINISH = 2)

*****
** Control file commands.

.CONTROL         = C1.QPL
.SAVECONTROLIF  (FINISH = 2)

.GET = CASE
.GET = NAME
.GET = COLLEGE
.GET = PLAYER
.GET = PHONE

*****
** Questions.

.QUESTION = INDATE, TYPE = XDATE
Date data was first entered.
.ANSWER
.NEXT

.QUESTION = INTIME, TYPE = TIME
Time of day data was first entered.
.ANSWER
.NEXT
```

Chapter 6 Scheduling Interviews

```
.QUESTION = TTL, TYPE = VOID, BORDER = RED

      QUESTIONNAIRE PROGRAMMING LANGUAGE
      Version 4.0

-- { Control File Demonstration I } --

      U.S. General Accounting Office
      July 29, 1991

.ANSWER
Press ENTER to begin...
.NEXT

.QUESTION = START, TYPE = MULT
Call the respondent:

      [NAME], [COLLEGE]
      [PHONE]

Did the respondent answer the phone?
.ANSWER
Yes, respondent answered
No, respondent did not answer
.GOTO = FINISH
.NEXT

.QUESTION = Q1, TYPE = STRING
Now ask the respondent the
first question...
.ANSWER = 20
.NEXT

.QUESTION = Q2, TYPE = NUMBER = "$##,###.##"
Now ask the respondent the
second question...
.ANSWER
.NEXT

.QUESTION = Q3, TYPE = CHECK
Now ask the respondent the
third question...
.ANSWER
First answer
Second answer
Third answer
Fourth answer
Fifth answer
Sixth answer
.NEXT

*****
** Questions that determine if the record
** will be saved and if the control file will
** be updated.
```

```
.QUESTION = FINISH, TYPE = MULT, BORDER = RED
.PUT = STATUS
Questionnaire completed.
You may now erase, review
or save this information.
.ANSWER = 2
Erase this interview
.GOTO = MAKESURE
Save Interview
Review questions
.GOTO = TTL
.NEXT = ENDTIME

.QUESTION = MAKESURE, TYPE = MULT, BACKGROUND = RED
Are you sure you want to erase
your answers?
.ANSWER = 2
Yes, erase these answers
No, do not erase these answers
.GOTO = FINISH
.NEXT

.QUESTION = ENDTIME, TYPE = TIME
Time data entry completed...
.ANSWER
.NEXT
```

Note: A copy of this program, Q1.DOC, has been included on the QPL Master Program Disk.

CONTROL Command

This example shows how a control file is linked to a questionnaire. Four new command words are used to do this: CONTROL, GET, PUT, and SAVECONTROLIF. The CONTROL command does two things. It tells the COLLECT and KEYPUNCH programs, first, that a control file is being used with this questionnaire and, second, what the name of the control file is. In this case, the control file is called "C1.QPL."

When the COLLECT or KEYPUNCH programs are started, they will first load the questionnaire, and then check to see if a control file is being used. If a control file is being used, they will also load that questionnaire program, then scan through the control file database selecting calls that have not been completed (according to the expression used in the control questionnaire's SHOWIF command), and then display the list of outstanding calls on the interviewer's screen.

GET Command

Once the interviewer selects one of the outstanding calls, the COLLECT and KEYPUNCH program will first load the entire record for that call from the control data file, and then proceed through the questionnaire. In the above example, five GET commands have been used to copy data from the control data file to the questionnaire. The variable names of questions in the control questionnaire are used to tell the COLLECT and KEYPUNCH programs what information you want to copy.

Example

Source Questionnaire Program	Questionnaire Format
...	...
.CONTROL = C1.QPL .SAVECONTROLIF (FINISH = 2)	
.GET = CASE	1 Enter unique case identification number. 1:1-4 * _ _ _ _
.GET = NAME	2. Enter respondent's name 1:5-34 * _____
.GET = COLLEGE	3. Enter college or country. 1:35-74 * _____
.GET = PLAYER	4. Enter shirt number 1:75-76 * _ _
.GET = PHONE	5. Enter respondent's phone number 2:1-10 * (_ _ _ _) _ _ _ _ - _ _ _ _
...	... SAVE IF (#13 = 2) SAVE CONTROL IF (#13 = 2)

The portion of the questionnaire program containing the GET commands has been repeated in this example to help clarify how they work.

GET commands are treated as if they were questions when your program is compiled. They tell the COMPILE program to copy the question text, type, and field size information for each of these questions from the control file program. When your questionnaire is printed with the CONVERT program, as shown on the right, they will appear as if they were typed in your source program. The asterisk printed to the right of the card and column location indicates that the data from this question will be copied from a control data file.

During the interview, the COLLECT and KEYPUNCH programs will automatically copy the information for each of these questions from the control data file as it proceeds down the list. If you use skip instructions to skip over a GET question, the data for that GET question will not be copied from the control data file. The COLLECT program does not display these questions on the screen; it simply

copies the answer and moves to the next question. The KEYPUNCH program displays GET questions, but it will not allow the interviewer to change them.²

You can refer to GET questions in your questionnaire program just as if you typed them yourself. That is, you may:

- skip to GET questions using GOTO, IF, SKIP, SKIPIF, or NEXT commands,
- use a GET question in expressions, and
- display the current answer of a GET question in the text of another question for the interviewer.

The questionnaire program shown above provides a number of examples of these uses.

GET questions, however, do have one limitation. You may not use a skip command with a GET question to change the skip path. The COLLECT and KEYPUNCH programs always will go to the next question in the list. You must use skip instructions with questions before or after a GET question to change the skip path. This restriction is generally not a problem because GET questions are typically used in a group, before the questionnaire gets underway. Later, after asking introductory questions, you could use an IF command to jump to a series of questions based on the answer to a GET question.

PUT Command

The PUT command provides a way copying information that the interviewer has entered in the questionnaire to the control data file. The most common use of the PUT command is to mark a respondent record to indicate that he or she has been called. Then, the next time the interview starts a questionnaire, that respondent's name will no longer appear in the selection list.

You may copy data only from matching types of questions in your questionnaire and control file. For example, the answer from a multiple choice question in the questionnaire may be copied only to a multiple choice question in your control file. Further, you must use the same field size for both questions. Finally, a PUT command must be used within the question it is copying (that is, after the QUESTION command and before the NEXT command).

²The KEYPUNCH program allows responses to GET questions to be edited when the record is loaded using one of the Retrieve Menu functions.

Example

Source Questionnaire Program	Questionnaire Format
...	
.SAVEIF (FINISH = 2) .SAVECONTROLIF(FINISH = 2)
.QUESTION = FINISH .TYPE = MULT .PUT = STATUS Questionnaire completed. You may now erase, save, or review this information. .ANSWER = 2 Erase this interview .GOTO = MAKESURE Save interview Review questions .GOTO = TTL .NEXT	13. Questionnaire completed. You may now review, save, or erase this information. (CHECK ONLY ONE ANSWER) __ 1. Erase this interview (GO TO QUESTION 14) __ 2. Save interview __ 3. Review questions (GO TO QUESTION 8) SAVE IF (#13 = 2) SAVE CONTROL IF (#13 = 2)

Parts of the questionnaire program example are shown above to highlight the PUT command.

In this example, the PUT command has been used with the FINISH question, which the interviewer uses to save or reject the interview record. The PUT command tells the COLLECT or KEYPUNCH program to copy the answer for this question to the question called STATUS in a respondent's control file record. The answer to FINISH will be copied to STATUS immediately after the interviewer enters a response.

Example

Source Control Program	Questionnaire Format
.SHOWIF (STATUS = 1) ...	SHOW IF (#8 = 1) ...
.QUESTION = STATUS TYPE = MULT Status of call .ANSWER Outstanding Complete .NEXT	6. Status of call __ 1. Outstanding __ 2. Complete

If the answer to STATUS is changed from one to two, then this respondent record will not be displayed (according to the SHOWIF command) the next time the interviewer starts a questionnaire.

SAVECONTROLIF Command

The SAVECONTROLIF command is used to tell the COLLECT and KEYPUNCH programs when the respondent's control data file record should be updated. After the interviewer passes the last question in the questionnaire, these programs will first evaluate the SAVEIF command's expression to determine if the interview record should be saved, and then they will evaluate the SAVECONTROLIF command's expression to determine if the control file record should be resaved. The control file record always will be updated if you do not use the SAVECONTROLIF command in your questionnaire program. In this example, the control file record will be updated when the interviewer selects the "Save interview" response to the FINISH question.

Since the interview record and the control file record may be saved independently, this allows you to program for situations where you need to make changes in the control file but not save a questionnaire record. For example, if you need to change control file information but have not started the interview, you can program your questionnaire so that it asks the interviewer for changes, then updates the control file for the next attempted call, but does not save an interview record. You could also add a question in the control file keeps track of the number of times a call has been attempted, while only saving an interview record when the call has been completed successfully. These programming techniques are discussed next.

Additional Control File
Commands

The following examples show how more elaborate control file and questionnaire programs could be written to allow control file information to be updated, incomplete interviews to be reloaded, and the number of attempted calls to be tracked. These two programs build upon the previous examples.

Sample Advanced Control File Program

```
*****  
** PROGRAM: C2.DOC                                DATE: 07/29/1991 **  
** AUTHOR: KEVIN DOOLEY                          **  
**                                               **  
** TITLE: QPL Control File Demonstration II      **  
**                                               **  
** PURPOSE: Demonstrate how to a CONTROL file to **  
**           restrict what cases may be entered. **  
**                                               **  
**           This program will create a data base of **  
**           respondents that a second program   **  
**           will use to control what cases the  **  
**           interviewer may enter. (In this example, **  
**           the data base will be made up of members of **  
**           the Washington Diplomats soccer team.) **  
**                                               **
```

Chapter 6 Scheduling Interviews

```
**          This control file is used by the questionnaire**
**          program: Q2.DOC. It demonstrates additional **
**          control file commands that may be used to   **
**          update control file information and restart **
**          incomplete interview records.                **
*****
** Global commands.

.TITLE      = "QPL Control File Demonstration II"
.SUBTITLE   = "Version 4.0"
.ESCAPE     = FINISH
.UPPERCASE  = OFF
.SAVEIF     (FINISH = 1)

*****
** Control file commands.

.HEADER     = "[NAME], [COLLEGE] -- [STATUS], [ATTEMPTS]"
.SHOWIF     (STATUS < 3 AND ATTEMPTS <= 25)
.IDENTIFY   = CASE
.RELOADIF   (STATUS = 2)

*****
** Questions for control file data base.

.QUESTION = CASE, TYPE = NUMBER
.LOW = 1
Enter unique case identification number.
.ANSWER = 4
.NEXT

.QUESTION = NAME, TYPE = STRING
Enter respondent's name.
.ANSWER = 30
.NEXT

.QUESTION = COLLEGE, TYPE = STRING
Enter college or country.
.ANSWER = 40
.NEXT

.QUESTION = PLAYER, TYPE = NUMBER
Enter shirt number.
.ANSWER = 2
.NEXT

.QUESTION = PHONE, TYPE = STRING = "(###)###-####"
Enter respondent's phone number.
.ANSWER
.NEXT

.QUESTION = STATUS, TYPE = MULT
Status of call
(Select 'Outstanding' when creating data base.)
.ANSWER
Outstanding
Incomplete
Complete
.NEXT
```

```
.QUESTION = ATTEMPTS, TYPE = NUMBER
Number of attempted calls
(Enter '0' when creating data base.)
.ANSWER = 2
.NEXT

*****
** Question used by SAVEIF to determine if data record
** should be saved.

.QUESTION = FINISH, TYPE = MULT
You may now save or erase this record.
.ANSWER
Save answers
Erase answers
.NEXT
```

Note: A copy of this program, C2.DOC, has been included on the QPL Master Program Disk.

Two new commands are used in this control file program, RELOADIF and IDENTIFY. These commands add the ability to reload existing interview records from the main data file back into the COLLECT or KEYPUNCH programs. This is usually done so that an interviewer can complete an interview that was interrupted.

RELOADIF Command

The RELOADIF and IDENTIFY commands work in concert with the SHOWIF command, which determines which cases will be displayed to the interviewer in the Control List. In this example, the SHOWIF command has been set so that only cases that are outstanding or incomplete (according to question STATUS), and have been called less than 26 times (according to question ATTEMPTS), will be shown. After the interviewer selects one of these cases, the COLLECT and KEYPUNCH programs check to see if the RELOADIF command was used. If it was not used, they begin the interview with a new record. If it was used, they evaluate its expression to determine if they should reload the interview record for the case the interviewer selected from the main data file.

IDENTIFY Command

The IDENTIFY command is used when a record needs to be reloaded from the main data file. It tells the COLLECT and KEYPUNCH programs which question holds a unique case identification number that can be used to find the matching interview record in the main data file. In the example above, question CASE is used to hold this number. The COLLECT and KEYPUNCH will search through the main data file until they find the matching record, then load it, and then begin the interview.

Several additional commands may be used in the questionnaire program to add even more flexibility to control files. The following sample program demonstrates how an incomplete interview can be automatically restarted at the point at which it was interrupted, how the number of attempted calls can be

counted, and how data from the control file may be edited and updated with or without saving an interview record. This example builds upon the previous questionnaire example.

**Questionnaire Program Using an
Advanced Control File**

```
*****
** PROGRAM: Q2.DOC                                DATE: 07/29/1991 **
** AUTHOR: KEVIN DOOLEY                          **
**                                               **
** TITLE: QPL Control File Demonstration II      **
**                                               **
** PURPOSE: Demonstrate how to a CONTROL file to **
**           determine what cases may be entered. **
**                                               **
**           This program uses a data base of respondent **
**           names and phone numbers to control who **
**           the interviewer may call that were created **
**           with control file program C2.DOC.     **
**                                               **
**           This version of the program uses additional **
**           control file features, including updating **
**           control file information, tracking the number **
**           of attempted calls, and reloading incomplete **
**           interviews.                          **
**                                               **
*****
** Global commands.

.TITLE           = "QPL Control File Demonstration II"
.SUBTITLE        = "Version 4.0"
.ESCAPE          = FINISH
.HEADER          = "[NAME], [COLLEGE]"
.UPPERCASE       = OFF
.SAVEIF          (FINISH > 1 AND START = 1)

*****
** Control file commands.

.CONTROL         = C2.QPL
.SAVECONTROLIF  (FINISH > 1 OR START > 1)

.IDENTIFY        = CASE
.PAUSE          = LASTQ
.INCREMENT       = ATTEMPTS

.GET = CASE
.GET = NAME
.GET = COLLEGE
.GET = PLAYER
.GET = PHONE
.GET = STATUS
```

Chapter 6 Scheduling Interviews

```
*****
** Questions.

.QUESTION = FIRSTD, TYPE = XDATE
Date data was first entered.
.ANSWER
.NEXT

.QUESTION = FIRSTT, TYPE = TIME
Time of day data was first entered.
.ANSWER
.NEXT

.QUESTION = LASTD, TYPE = DATE
Date of last call.
.ANSWER
.COMPUTE (DATE - 19000000)
.NEXT

.QUESTION = LASTT, TYPE = TIME
Time of last call.
.ANSWER
.COMPUTE (TIME)
.NEXT

.QUESTION = TTL, TYPE = VOID, BORDER = RED

    QUESTIONNAIRE PROGRAMMING LANGUAGE
        Version 4.0

-- { Control File Demonstration II } --

    U.S. General Accounting Office
        July 29, 1991

.ANSWER
Press ENTER to begin...
.NEXT

.QUESTION = START, TYPE = MULT
Call the respondent:

    [NAME], [COLLEGE]
    [PHONE]

Did the respondent answer the phone?
.ANSWER
Yes, respondent answered
No, could not get through
.GOTO = ENDTIME
No, phone number was incorrect
.GOTO = NEWPHONE
.NEXT = Q1

*****
** Update control file data.

.QUESTION = NEWPHONE, TYPE = STRING
What is the correct phone number?
.ANSWER = PHONE, COPY = PHONE, PUT = PHONE
.NEXT = ENDTIME
```

Chapter 6 Scheduling Interviews

```
*****
** Continue questions.

.QUESTION = Q1, TYPE = STRING
Now ask the respondent the
first question...
.ANSWER = 20
.NEXT

.QUESTION = Q2, TYPE = NUMBER = "$##,###.##"
Now ask the respondent the
second question...
.ANSWER
.NEXT

.QUESTION = Q3, TYPE = CHECK
Now ask the respondent the
third question...
.ANSWER
First answer
Second answer
Third answer
Fourth answer
Fifth answer
Sixth answer
.NEXT

*****
** Questions that determine whether the interview will be
** saved and control file will be updated.

.QUESTION = FINISH, TYPE = MULT, BORDER = RED
Questionnaire completed.
You may now review, save or erase
this information.
.ANSWER = 3
Erase this interview
.GOTO = MAKESURE
Save incomplete interview
Save complete interview
Review questions
.GOTO = TTL
.NEXT = ENDTIME

.QUESTION = MAKESURE, TYPE = MULT, BACKGROUND = RED
Are you sure you want to erase
your answers?
.ANSWER
Yes, erase these answers
No, do not erase these answers
.GOTO = FINISH
.NEXT

.QUESTION = ENDTIME, TYPE = TIME
Time first data entry completed...
.ANSWER
.NEXT
```

```
.QUESTION = STATUS2, TYPE = MULT
Compute current interview status
.ANSWER
Outstanding
Incomplete
Complete
.COMPUTE (FINISH)
.COMPUTEIF (START > 1) (STATUS) * use previous status setting
.PUT = STATUS
.NEXT

.QUESTION = LASTQ, TYPE = NUMBER
Number of last question displayed before
the Escape key was pressed.
.ANSWER = 3
.COMPUTE (ESCAPE)
.NEXT
```

Note: A copy of this program, Q2.DOC, has been included on the QPL Master Program Disk.

Several new commands are used in this version of the questionnaire program example: IDENTIFY, PAUSE, INCREMENT, and COPY. Several new questions also have been added to hold information on the unique case identification number (CASE), an edited version of a control file question (NEWPHONE), the status of the case (STATUS2), and the number of the last question that was displayed before the interviewer pressed the Escape key (LASTQ).

IDENTIFY Command

The IDENTIFY command must be used in the questionnaire program when you use the RELOADIF command in the control file program to reload incomplete interview records. This tells the COLLECT and KEYPUNCH programs how to match the control file record the interviewer chose with the correct interview record in the main data file. In this example, a GET command was used to automatically copy the unique record identification number from question CASE to the questionnaire program. This guarantees that the two records will have matching identification numbers. You instead could have the interviewer enter the case number, but this would allow a greater opportunity for an error to occur.

PAUSE Command

The PAUSE command tells the COLLECT and KEYPUNCH programs which question will be used to hold the number of the question that was displayed on the screen immediately before the interviewer pressed the Escape key to end an interview early. This information will be used later when the interviewer reloads an incomplete interview. In this case, the COLLECT and KEYPUNCH programs will automatically start the questionnaire at the last question that was displayed. If the PAUSE command is not used, or if the interviewer never pressed the Escape key, the reloaded interview will be started at the beginning.

In the above sample program, question LASTQ is used to hold the number of the last question displayed. A COMPUTE command is used with the ESCAPE keyword

to get this question number from the COLLECT or KEYPUNCH programs and use it as an answer for question LASTQ. The ESCAPE keyword always has a value of the number of the question on the screen when the Escape key was pressed. If the Escape key has not been pressed, the ESCAPE keyword will have a value of zero. Also notice that this question was placed after question FINISH, the question the interviewer skips to after pressing the Escape key (according to the ESCAPE command). This insures that the answer to LASTQ will be computed after the interviewer has pressed the Escape key and has begun exiting the questionnaire.

INCREMENT Command

The INCREMENT command is used to keep track of the number of times an interviewer tried to call a respondent. In the above sample program, it tells the COLLECT and KEYPUNCH programs to increase the value of question ATTEMPTS (in the control file record) by one each time the control file record is updated (according to the SAVECONTROLIF command). Also note that this value is used in the control file program's SHOWIF command to limit the number of times that an interviewer can call a respondent.

Only one INCREMENT command may be used in a questionnaire program to increment an answer to a question. Also, only number questions may be used with the INCREMENT command.

COPY Command

The COPY command is generally used to edit information in the control file record while using the questionnaire program. In the sample program above, it has been used to copy the telephone number from question PHONE (where it was copied from the control file using the GET command) to question NEWPHONE where it may be edited. The PUT command is used in question NEWPHONE to move the revised telephone number back to the control file record. Then, the next time the interviewer selects this case from the Control List, the revised telephone number will be displayed.

The COPY command may be used to copy the current answers to any types of questions, but both questions must be the same type and have the same field size. In the sample program, these requirements were guaranteed because NEWPHONE's ANSWER command was set to PHONE, which automatically copies its field size.

The COPY command copies an answer from another question only the first time an interviewer sees a question. If the interviewer backs up to the question or reloads it (either as an incomplete interview using the Control List or using one of the KEYPUNCH Retrieve Menu functions), the answer from the other question will not be recopied.

Importing Control Data Files

Respondent data files from other sources may be used as control data files. For example, if you already have a database of respondent names and phone numbers, you can use it to build a QPL control data file. If your data file is in a fixed format, the process of using it with a QPL control file is relatively easy. In this case, you only need to write a QPL program that exactly matches the record layout of that data file.

If the data is not in a fixed format, you must write a program that will translate the data into this format. How you do this depends on how the data are formatted. For example, a dBase DBF file can be translated to a fixed format file using the dBase “COPY TO filename.DAT SDF” command. Lotus 1-2-3 files can be translated by printing the data using the “/Print File” command. If the data are in another format, you may have to write a program in a general purpose language, such as BASIC, to translate it. After you have translated the data, you must write QPL control file program that will read that data file.

If you need to add new variables to your control program, such as questions that specify when a call is outstanding or completed, you can create a new version of your control file program, including new questions and dropping some you do not need. Then you can use the QPL CONVERT program to automatically translate the data file from the original record layout to the record layout of your modified control program and then use the KEYPUNCH program to edit the new fields.³

³See chapter 12.

Foreign Languages

QPL programs can be set up for CATI interviews in any language that can be represented in the IBM/PC extended character set. The American-style formats for numbers and dates that are used in the COLLECT and KEYPUNCH programs can be changed to the customary formats of another country, and the error and other messages they display to the interviewer also may be translated. The following changes must be made in order to run a CATI in another language:

- Specify your country using the DOS COUNTRY command in your CONFIG.SYS file.
- Set the keyboard character set using DOS KEYB command.
- Translate the QPL error messages, which are listed in the MESSAGE.DOC file on the Master Program Disk, to your language.
- Write your questionnaire program in your language (using the INCLUDE command to add the translated error messages).

The data formatting information and the error messages will be stored with your questions in the compiled questionnaire (.QPL) file. Thus, your interviewers' computers need only your compiled questionnaire program and the DOS KEYB command set to your language to operate properly. You do not need to be concerned about the COUNTRY setting in their CONFIG.SYS files.

Reconfiguring Your Computer

Each time you start your computer, it always looks for a file called CONFIG.SYS. This file tells DOS how you want your computer configured. Among other things, it tells DOS what country the computer is being used in. With this information, DOS can print the time, date, and other information in the format that is customary in that country. For example, most European countries write dates beginning with the day, followed by the month and the year (*i.e.*, “9 October 1987”). In the United States, dates customarily begin with the month, followed by the day and year (*i.e.*, “October 9, 1987”).

QPL programs also use this information to configure themselves to the local customs for formatting dates and times. For example, if the country is set to France in the CONFIG.SYS file, the COLLECT and KEYPUNCH programs will automatically display date questions in day-month-year format. The date, however, will still be stored in YYMMDD format in the data file.

QPL programs also will follow the local conventions when formatting numbers. For example, if the country is set to France in the CONFIG.SYS file, the QPL programs will recognize a comma as a decimal point in a number format string instead of a period, and a space as a thousands separator instead of a comma.

Example

Source Program	Questionnaire Format
.QUESTION = Q1	1. ...
.NUMBER = "### ##,##FF"	...
...	...
...	_ _ _ _ _ _ _ _ , _ _ _ _ FF
.ANSWER	
.NEXT	

In this example, question Q1 is formatted for French Francs. It is an eight-digit number, including two decimal places. The COMPILE program will automatically use the comma to calculate the number of decimal places, and the CONVERT program will correctly specify the number of decimal places when it creates an SPSS or SAS program, or translates the data to Lotus or dBase files.

COUNTRY Command (DOS)

The DOS COUNTRY command must be used in your CONFIG.SYS file before the computer is turned on. Thus, you must edit the CONFIG.SYS file, adding the country specification, save the edited file, and then turn the computer off. When you restart the computer, DOS will read the CONFIG.SYS file and configure itself to that country.

It is not necessary, however, to configure the computers that your interviewers will be using to a particular country. The COMPILE program stores the country-specific information from your computer in the compiled questionnaire file. The COLLECT and KEYPUNCH programs will use this information, instead of the settings of the particular computer it is running on, to determine how to display dates, times, and numbers.

KEYB Command (DOS)

The DOS KEYB program can be used to convert your keyboard into a keyboard of another country. This allows you to use characters that are not used in American English. The keyboard must be set correctly both when you are typing your program and when your interviewers are using your questionnaire program. The KEYB command may be used to set the keyboard in your CONFIG.SYS file, typed at the DOS command line, or included in your AUTOEXEC.BAT file. The following example shows how it and the COUNTRY command could be used in a CONFIG.SYS file.

Sample CONFIG.SYS File

```
FILES= 30
BUFFERS=20
BREAK=ON
SHELL=C:\DOS\COMMAND.COM C:\DOS\ /E:256 /P
DEVICE=C:\ANSI.SYS
COUNTRY=003, ,C:\DOS\COUNTRY.SYS
INSTALL=C:\DOS\KEYB.COM LA, ,C:\DOS\KEYBOARD.SYS
```

In this sample file, the country will be set to Latin America (country code 003), and the keyboard also will be set to Latin America (keyboard code LA). Once the keyboard has been set, you may switch back to the U.S. keyboard by pressing Ctrl-Alt-F1. You may switch to your country's keyboard again by pressing Ctrl-Alt-F2. The country-specific data formatting information, however, cannot be changed unless you reboot your computer with a new country setting in your CONFIG.SYS file. See your DOS manual for more information on these commands.

Translating QPL Error Messages

All of the messages the COLLECT program displays to the interviewer, such as error messages, help screens, and phrases used in dialogue boxes, may be translated to other languages. Many of the messages used in the KEYPUNCH and CONVERT programs also may be translated. To translate a phrase, you must set a message's code number equal to the new phrase. The phrase must be used on a command line, and it must be delimited with quotation marks.

Translation Syntax

```
.code number = "new phrase"
```

The QPL programs have over 200 messages that may be translated. The English version of these messages, and their code numbers, are listed in the file called MESSAGE.DOC, which has been included on the QPL Master Program Disk.

Use the English message in this file as a guide for the maximum length of the translation. While there is some flexibility, no message may be more than 76 characters long and many should not be longer than one word. The COMPILE program does very little error checking on message sizes, so you should test your questionnaire program to ensure that your message will be displayed properly.

Sample SPANISH.DOC Program

```
* 12 = "QUESTION"
. 12 = "PREGUNTA"

* 13 = "Not known"
. 13 = "No se sabe"

* 16 = "Invalid date."
. 16 = "La fecha no es válida."

* 19 = "Number may not be lower than" * number
. 19 = "Los números no deben ser más bajos que"

* 20 = "Number may not be higher than" * number
. 20 = "Los números no deben ser más altos que"

* 21 = "Answer may not be less than" * string
. 21 = "La respuesta no debe ser más baja que"

* 22 = "Answer may not be greater than" * string
. 22 = "Las respuesta no debe ser más alta que"
```

This example shows how you could write a QPL program to change some of the default phrases and error messages to Spanish. Note that the original English messages are used as comments since their lines begin with an asterisk. The Spanish translations follow each English message on a command line, which begins with a period.

INCLUDE Command

Generally, you should keep your translated error message program file separate from your source questionnaire program files. You can let the COMPILE program automatically merge your message program into your questionnaire program by using the INCLUDE command. This command tells the COMPILE program to stop reading your source questionnaire program and begin reading the file you have specified. When it has finished reading that file, it will continue reading your source questionnaire program again. This allows you to use your message file with any questionnaire project without modifying it.

Sample PROG1.DOC Program

Source Program	Questionnaire Format
.INCLUDE = SPANISH.DOC	
.QUESTION = Q1, TYPE = MULT	1. ...
...	...
...	
.ANSWER	(Marque sólo una respuesta.)
Sí	
No	__ 1. Sí
No se sabe	__ 2. No
.NEXT	__ 3. No se sabe

In this example, the INCLUDE command was used to merge the message translation file, called SPANISH.DOC, into the questionnaire program. You could include the disk drive and path name for this file if it is different from the source questionnaire program. Note that the column on the right shows that the “Check only one answer” phrase has been replaced with the equivalent Spanish phrase when the program is printed in questionnaire format.¹

Since the data formatting information and the translated messages are included in the compile questionnaire file, all that remains is to make sure that the interviewer is using the correct keyboard. The following program demonstrates how a DOS batch file could be used to set the keyboard and start the COLLECT program with your questionnaire.

Sample GO.BAT Batch File

```
KEYB LA
COLLECT PROG1.QPL
KEYB US
```

In this example, a batch file called GO.BAT will use the DOS KEYB command to switch to the Latin American keyboard and then it will start the COLLECT program with the questionnaire program called PROG1.QPL. After the interviewer finishes using the COLLECT program, the second KEYB command will switch the keyboard back to the United States. See chapter 9 for more information on writing batch files to start the COLLECT program.

¹A copy of the SPANISH.DOC message file has been included on the QPL Master Program Disk. It contains translations for the more commonly used COLLECT program and CONVERT questionnaire document messages.

Running the COMPILE Program

The COMPILE program reads your questionnaire program and creates a shortened, or “compiled,” version of the questionnaire. This file will have the same name as your questionnaire program (and be written to the same disk drive and directory), but the extension will be changed to “QPL.”

This version of your questionnaire is used by the other QPL system programs: COLLECT, NCOLLECT, KEYPUNCH, and CONVERT. If COMPILE finds an error in your program, such as incorrect syntax or missing questions, it will print an error message on the screen and in an error log file, and continue scanning your program. Your program will not be compiled if an error is found.¹

There are two ways to tell the COMPILE program which file you would like to process. The easiest way is to use the “Startup Menu,” which asks you for the name of your questionnaire and control programs, as well as any options you may wish to use.

Startup Menu

```

Files
  Source: C:\
  Control: C:\

Options
  [ ] Black and white screen
  [ ] Display source code
  [ ] Extended error messages

BEGIN  Cancel
  
```

The Startup Menu is displayed when you start the COMPILE program without any options, that is, when you start it by typing “COMPILE,” and then pressing the Enter key.

Example

```
C:\> COMPILE
```

The Startup Menu lets you edit the file names and options before you start the program. A highlight bar indicates which item in the menu you are pointing at. You can move the highlight bar up and down using the arrow keys on the right side of your keyboard or by pressing the Enter key.

¹See appendix I for instructions on how to install the QPL programs on your computer.

The highlight bar always points to the source file name when you first start the program. You must type the name of the source questionnaire file you want to process in this space. Entering a control file name is optional. You may use the insert, delete, and backspace keys to edit file names.

Press the space bar to toggle the options on and off. A check mark shows when an option is on. How the options work is discussed below.

You must press the Enter key to jump from the options to the “Begin” box. Press Enter at this point to start the program, or use the arrow keys to highlight “Cancel” to quit the program. Pressing the up arrow key will take you back to the file names and options. You may also press the Escape key at any time to quit the Startup Menu and return to DOS.

The second way to specify your files and options is to type them at the DOS command line when you start the COMPILE program. Although you may find this method to be more cumbersome to use at first, you probably will find it to be faster as you get used to using the COMPILE program, especially if you write a DOS batch file to start it with your files and options.

DOS Command Line Options

```
COMPILE [sourcefile.ext...] [/B] [/Ccontrolfile.QPL] [/D]
        [/Errorlog.ERR] [/H] [/M] [/N] [/Oobjectfile.QPL] [/P#]
```

```
/B Set screen colors to black and white
/C Set control file name
/D Display source files on the screen
/E Rename error log file
/H Display command line options
/M Show all error messages (default is first error per question)
/N Do not pause after compiling
/O Rename QPL output file
/P# Set password number (range 1 to 32767)
```

Note: The brackets indicate optional information that may be included when starting the program. The uppercase letters indicate the names of the options, and the lowercase letters indicate type of argument that may be used by an option. The “#” symbol indicates that a number is required. Do not type any spaces between the number and the option letter.

As you may have already noticed, more options are available at the DOS command line than the Startup Menu. These additional options, however, are more rarely used.

Multiple Source Files

You may specify more than one source questionnaire file name if you start the COMPILE program at the DOS command line. This feature is useful when you are developing several variations of one questionnaire. Questions that do not change among different versions of the questionnaire can be kept in files separate from questions that are unique to each version. The COMPILE program will process each of the files in the list and merge them into one QPL program. The name of the first source file in the list will be used to name the QPL file. Up to 10 source file names may be specified at the command line.

Example

```
C:\> COMPILE A:SAMPLE.DOC A:SECTION1.DOC A:SECTION2.DOC
```

In this example, three source questionnaire files have been specified when the COMPILE program was started. The COMPILE program will first read SAMPLE.DOC, then SECTION1.DOC, and finally SECTION2.DOC. If there are no errors in any of the programs, the COMPILE program will create a QPL file called SAMPLE.QPL.

INCLUDE Command

There is an alternative way of merging two or more source questionnaire files into one program. You can use the INCLUDE command in your questionnaire program to tell the COMPILE program to read other source questionnaire files. This command is often used to include a program that contains error messages for the COLLECT program that have been translated to another language.²

The example below shows how this command could be used to combine the same three files into one QPL program. In this case, only the first program (A:SAMPLE.DOC) needs to be specified in the Startup Menu or at the command line. The COMPILE program will read the other two programs (A:SECTION1.DOC and A:SECTION2.DOC) when it processes each INCLUDE command. It will resume processing the original source program after it reads everything in each included program.

²See chapter 7.

Example

```
Source Questionnaire Programs      Questionnaire Format

File: A:SECTION1.DOC

.QUESTION = Q2, TYPE = MULT
(Question from A:SECTION1.DOC)
.ANSWER = Q1
.NEXT

-----

File: A:SECTION2.DOC

.QUESTION = Q3, TYPE = MULT
(Question from A:SECTION2.DOC)
.ANSWER = Q1
.NEXT

-----

File: A:SAMPLE.DOC                File: A:SAMPLE.TXT

.QUESTION = Q1, TYPE = MULT        1. (Question from A:SAMPLE.DOC)
(Question from A:SAMPLE.DOC)      |__| 1. Yes
.ANSWER                            |__| 2. No
Yes                                |__| 3. Don't know
No
Don't know
.NEXT

.INCLUDE = A:SECTION1.DOC         2. (Question from A:SECTION1.DOC)

                                   |__| 1. Yes
                                   |__| 2. No
                                   |__| 3. Don't know

.INCLUDE = A:SECTION2.DOC         3. (Question from A:SECTION2.DOC)

                                   |__| 1. Yes
                                   |__| 2. No
                                   |__| 3. Don't know
```

In this example, questions Q2 and Q3 are stored in separate source files, called SECTION1.DOC and SECTION2.DOC. They will be inserted into the main questionnaire source file, called SAMPLE.DOC, at the insert commands. Since questions Q2 and Q3 are inserted after question Q1, it is allowable for these two questions to copy the answer list from question Q1. The column on the right shows how this program will be printed in questionnaire format by the CONVERT program.

Any number of source questionnaire files may be merged using INCLUDE commands. Further, you may nest INCLUDE files, that is, one included file may include another source questionnaire file. The number of files specified in your CONFIG.SYS limits how deeply you may nest files.

Options

As noted above, there are a number of options you may select when starting the COMPILE program to modify how it works. Three options are available from the Startup Menu, and eight options are available at the DOS command line. The additional command line options are more rarely used. If you use an option at the command line, you must always precede the option letter with a slash, and all of the options and file names you use must be separated on the line by at least one space.

Option /B, Set Screen Colors to Black and White

The COMPILE program displays the program title and error messages on your display screen in color if your computer has a color graphics display card.³ It automatically switches to black and white if your computer has a monochrome display card, such as a Hercules graphics card. Some computers, however, have a color graphics card connected to a monochrome monitor (IBM/PC portable computers and many laptop computers are configured this way). In this instance, the COMPILE program would use colors, but they may be unreadable on the monochrome screen. You can force the COMPILE program to use black and white by adding the /B option to the end of the command line when starting the program, or checking “Black and white screen” on the Startup Menu.

All the QPL programs (COMPILE, COLLECT, KEYPUNCH, and CONVERT) will use a black and white screen if you start them with the /B option.

Option /D, Display Source Code

This option tells the COMPILE program to print your source questionnaire on the screen as it is being processed. This makes it easier to see where an error may have occurred.⁴

DISPLAY Command

Displaying your source questionnaire program on the screen, however, greatly increases the time it takes the COMPILE program to process your questionnaire. You can minimize this problem if you tell the COMPILE program to show only parts of your program, using the DISPLAY command.

³CGA, EGA, or VGA compatible graphic display card.

⁴You can make the COMPILE program pause, so you can read the error message, by holding the Control (Ctrl) key down and pressing the Number Lock (NumLock) key, or by pressing the Pause key if you have one on your keyboard. Press any other key to resume processing the questionnaire.

Example

```
.QUESTION = Q1, TYPE = NUMBER
...
.ANSWER = 4
.NEXT

.DISPLAY = ON

.QUESTION = Q2, TYPE = DATE
...
.ANSWER
.NEXT

.DISPLAY = OFF

.QUESTION = Q3, TYPE = TIME
...
.ANSWER
.NEXT
```

This example shows how the DISPLAY command is used. When it is set to ON, the COMPILE program will show the source questionnaire program on the screen as it is being processed. It will stop when the display command is set to OFF, or when it reaches the end of your program. Here, only question Q2 will be displayed on the screen.

Option /M, Show All Error Mes-
sages

As discussed above, the COMPILE program normally shows only the first error it discovers within a question on the screen. Since the first error often will cause other errors, it is generally not useful to see all of them. You can, however, force it to show all of the errors it discovers using the /M option or by checking “Show all error messages” on the Setup Menu.

VERBOSE Command

You also may tell the COMPILE program to display all of the error messages by using the VERBOSE command in your source questionnaire program.

Example

```
.QUESTION = Q1, TYPE = TIME
...
.ANSWER
.NEXT

.VERBOSE = ON

.QUESTION = Q2, TYPE = DATE
...
.ANSWER
.NEXT

.VERBOSE = OFF
```

Chapter 8 Running the COMPILE Program

```
.QUESTION = Q3, TYPE = NUMBER
...
.ANSWER = 5
.NEXT
```

This example demonstrates how the `VERBOSE` command could be used to display all error messages for just one part of a questionnaire program. It is set to `ON` after question Q1, then it is set to `OFF` after question Q2. This tells the `COMPILE` program to display all possible error messages while it is processing question Q2. It will display only the first error it may find while processing questions Q1 and Q3.

Option /C, Set Control File Name

This option allows you to specify the name of a control file when you start the `COMPILE` program. Control files are QPL programs that are used to maintain a database of respondents.⁵ This option is equivalent to using the `CONTROL` command in your source questionnaire program. You may find it easier to set the control file at the DOS command line or in the Setup Menu when you are creating several versions of your questionnaire. For example, if you are customizing your questionnaire for each of your interviewers (such as giving each interviewer a unique questionnaire file name, password number, and control data file), it would be faster to use command line options (typically using a batch file) rather than changing the source questionnaire file.

DOS Batch File (RUN.BAT)

```
ECHO OFF
CLS
ECHO Creating four versions of SAMPLE.DOC questionnaire...
PAUSE
COMPILE SAMPLE.DOC /OSAMPLE1.QPL /ESAMPLE1.ERR /CLIST1.QPL /P111 /N
IF ERRORLEVEL 1 GOTO STOP
COMPILE SAMPLE.DOC /OSAMPLE2.QPL /ESAMPLE2.ERR /CLIST2.QPL /P222 /N
COMPILE SAMPLE.DOC /OSAMPLE3.QPL /ESAMPLE3.ERR /CLIST3.QPL /P333 /N
COMPILE SAMPLE.DOC /OSAMPLE4.QPL /ESAMPLE4.ERR /CLIST4.QPL /P444 /N
GOTO DONE
:STOP
ECHO *** ERROR - QPL files could not be created. ***
:DONE
```

Result:

```
SAMPLE1.QPL
SAMPLE1.ERR
SAMPLE2.QPL
SAMPLE2.ERR
SAMPLE3.QPL
SAMPLE3.ERR
SAMPLE4.QPL
SAMPLE4.ERR
```

⁵See chapter 6.

The batch file in this example shows how one questionnaire program could be customized for four different interviewers. The same source questionnaire file is used (SAMPLE.DOC), but command line options are used to give each QPL file, error log, and control files unique names and password numbers. The batch file also tests for any errors that may have occurred during the first compilation. Since the COMPILE program returns the number of errors to DOS, the batch file can use this information to print an error message and then stop processing files.

Option /E, Rename Error Log File

The COMPILE program normally creates an error log file that has the same name as your questionnaire program, and is written to the same disk drive and directory, but uses an "ERR" extension to indicate what type of file it is. You can tell the COMPILE program to use another name using the /E option. The new file name, including the disk drive letter and pathway information, must be typed immediately after the option letter on the command line. The above example shows how this option could be used in a batch file to rename error log files.

Option /N, Do Not Pause

The COMPILE program pauses after it finishes processing your questionnaire program and displays a message that says whether it was able to create a QPL file. You must press a key to clear this message and return to the DOS command line. The /N option tells the COMPILE program to immediately return to the DOS command line after it processes your program. This is useful when you will be using a batch file to compile several programs at one time, as in the above example.

Option /O, Rename QPL Output File

The COMPILE program uses the name of your source questionnaire program (or the first program if you are merging several) as the name for the QPL file, but changes the extension to QPL. You can use the /O option to tell the COMPILE program to use another name. The above example shows how this option is used.

Option /P#, Set Password Number

You can control who may use your questionnaire program with the COLLECT or KEYPUNCH programs (to a limited extent) by specifying a password that interviewer must enter in order to start the program. The password must be a number from 1 to 32,767. You can use the /P option to set the password number when you compile your source questionnaire program. The number must be typed on the DOS command line immediately following the option letter. The above example shows how this option could be used in a batch file to give each QPL file different password numbers.

PASSWORD Command

A password number may instead be set in your source questionnaire program using the PASSWORD command. Again, the password must be a number between 1 and 32,767.

Example

```
.PASSWORD = 342

.QUESTION = Q1, TYPE = NUMBER
  ...
.ANSWER = 5
.NEXT
```

The `PASSWORD` command can be used on any command line in your source questionnaire program, but it is usually used near the beginning. In this example, the password will be set to 342 when the `COMPILE` program processes the program.

**Option /H, Display
Command Line Options**

You can tell the `COMPILE` program to list all of the command line options on the screen if you start it with the `/H` option. Once the list is displayed, you can press any key to quit the `COMPILE` program and return to DOS. All the QPL programs (`COMPILE`, `COLLECT`, `KEYPUNCH`, and `CONVERT`) will display their command line options if you start them with the `/H` option.

Errors

Error messages may be displayed on the screen during any stage of the compilation. The line containing the error, and its line number, will also be displayed and copied to the error log file.⁶

An error in your program may not be at the line that the `COMPILE` program displays. It cannot check for many errors until it reaches the `NEXT` command in a question. Thus, the error may be in one of the lines just above the line that the `COMPILE` program listed.

Normally, the `COMPILE` program will display only the first error it discovers within a single question. This default is used because the first error often causes several other errors within the same question. You can, however, tell the `COMPILE` program to display all the errors it finds by checking the “Extended error messages” option on the Startup Menu or by using the `/M` option at the DOS command line.

The `COMPILE` program generates three types of error messages: warnings, errors, and fatal errors. It also displays a count of the number of errors and warnings at the bottom of the screen with the line and question number that it is currently processing.

⁶The DOS `EDIT` text editor is handy for correcting errors in your program because the line numbers it displays match the line numbers given in the `COMPILE` error messages.

Warnings	“Warnings” are recoverable errors. They are usually due to some limitation being exceeded, such as the length of a text line. In these situations, the COMPILE program will substitute the default specification and continue scanning your program. A program will be compiled even if warnings are given.
Errors	“Errors” are nonrecoverable errors. They are usually due to a command word being misspelled or a skip to a nonexistent question. The COMPILE program will continue to scan your program for more errors, but it will not create the compiled QPL file.
Fatal Errors	“Fatal errors” force the COMPILE program to stop immediately. They are usually caused by a memory limitation being exceeded, a disk drive problem, or a missing file.
Error Log File	The COMPILE program will display any errors it finds in your questionnaire program on the screen and it will copy them to an error log file. This file is automatically given the same name as your questionnaire program, but it is given a new extension, “ERR,” to identify it as an error log file. It is a simple text file that may be loaded into your word processor or printed using the DOS PRINT command.

Example

```
C:\> PRINT A:SAMPLE.ERR
```

The error log file will keep track of when the file was compiled, what files were used, where any errors may have occurred, and what the errors were.

Error Log File (A:SAMPLE.ERR)

```
Questionnaire Programming Language - Version 4.0
Error Log File
A:\SAMPLE.DOC
Mon Aug 05 12:20:11 1991

--> A:\SAMPLE.DOC

--> A:\SECTION1.DOC

<-- A:\SECTION1.DOC

--> A:\SECTION2.DOC

<-- A:\SECTION2.DOC

<-- A:\SAMPLE.DOC
```

```
Questions: 3
Record size: 3 bytes
```

```
No errors were detected.
```

This example shows the error log file for the SAMPLE.DOC program. The arrows indicate when a file is opened and closed for processing. Right arrows show when a file is open, and left arrows show when it is closed. Error messages that are listed just after a file is opened are usually due to syntax errors in that file. Error messages listed after the last file is closed are usually due to skips to undefined questions.

Stopping the COMPILE Program

When the COMPILE program finishes processing your program, it will display a message on the screen that says whether it was able to create the QPL file. Pressing any key at this point will exit the program and return you to the DOS command line. You can stop the program at any point by pressing the Escape key. The QPL file will not be created if you break out of the program early.

Running the COLLECT Program

The COLLECT program turns your questionnaire into a computer-aided telephone interview (CATI) program. It reads your compiled questionnaire and then displays each question, one at a time, on the computer screen. After all of the questions have been answered, it adds the record of the interview to the end of a fixed format data file. If the data file already exists, the COLLECT program will append the record to the end of the file. If the data file does not exist, it will open a new file. The COLLECT program has many features that are designed to help the interviewer correctly record a respondent's answers.

Instructions on operating the COLLECT program are given in the interviewer's manual, called the *QPL Data Collection Program—Version 4.0*. This chapter discusses the various options that may be used when starting the program and suggests how the programs and data files should be organized on the interviewer's disks.

There are two ways to tell the COLLECT program which questionnaire you want to use. One way is to use the "Startup Menu," which asks you for the name of your questionnaire and back up data file, and any options you may want to use.

Startup Menu

```

Files
  Source: C:\
  Back Up: C:\

Options
  [ ] Black and white screen
  [ ] Turn error beep off
  [ ] Use upper and lower case letters

BEGIN  Cancel
  
```

The Startup Menu is displayed when you start the COLLECT program without any options, that is, when you start it by typing "COLLECT" and then pressing the Enter key.

Example

```
C:\> COLLECT
```

The Startup Menu lets you edit the file names and options before you start the program. A highlight bar indicates which item in the menu you are pointing at.

You can move the highlight bar up and down using the arrow keys on the right side of your keyboard or by pressing the Enter key.

The highlight bar always points to the source file name when you first start the program. You must type the name of your compiled questionnaire file in this space. Typing the QPL extension, however, is optional. The COLLECT program will always load the file with this extension, even if you enter a file name with another extension. Entering a backup data file name is optional. Use the Insert, Delete, and backspace keys to edit file names.

Press the spacebar to toggle the options on or off. A check mark indicates when an option is on. These options are described below.

You must press the Enter key to jump from the options to the “Begin” box. Press Enter at this point to start the program, or use the arrow keys to highlight “Cancel” to quit the program. Pressing the up arrow key will take you back to the file names and options. You may also press the Escape key at any time to quit the Startup Menu and return to DOS.

You may instead start the COLLECT program with your questionnaire if you specify its file name at the DOS command line.

DOS Command Line Options

```
COLLECT [filename[.QPL]] [/B] [/E] [/Fbackupfile.ext]
        [/H] [/O#] [/P#] [/Q] [/R] [/T] [/U] [/Wname]
```

```
/B      Set screen colors to black and white
/E      Set error beep off
/F      Specify backup data file name
/H      Display command line options
/O#     Set OPTION keyword number
/P#     Enter password number (range 1 to 32767)
/Q      Do not display question number
/R      Do not restart interview
/T      Do not display program title
/U      Set uppercase control off
/Wname  Set workstation name
```

Note: The brackets indicate optional information. The lowercase words indicate the type of information required.

Example

```
C:\> COLLECT A:SAMPLE
```

In this example, the COLLECT program will use the questionnaire program, "SAMPLE.QPL," which is located in the A: disk drive. Note that typing the QPL extension is optional.

Options

As noted above, there are a number of options you may select when starting the COLLECT program that modifies how it works. Three options are available from the Startup Menu, and seven options are available at the DOS command line. The additional command line options are more rarely used. If you use an option at the command line, you must always precede the option letter with a slash, and all of the options and file names you use must be separated on the line by at least one space.

Option /B, Set Screen Colors to Black and White

The COLLECT program displays your questions on your display screen in color if your computer has a color graphics display card. It automatically switches to black and white if your computer has a monochrome display card, such as a Hercules graphics card. Some computers, however, have a color graphics card connected to a monochrome monitor (many laptop computers are configured this way). In this instance, the COLLECT program would use colors, but they may be unreadable on the monochrome screen. You can force the COLLECT program to use black and white by adding the /B option to the end of the command line when starting the program, or checking "Black and white screen," at the Startup Menu.

All of the QPL programs will use black and white if you start them with the /B option.

Option /E, Turn Error Beep Off

The COLLECT program will beep and display a message on the bottom line of the monitor screen if the interviewer makes an error, such as trying to enter a letter into a numeric question or entering a value that is too high or too low. You can turn this beep off by checking "Turn error beep off" on the Startup Menu or by using the /E option when starting the COLLECT at the DOS command line. Error messages, however, will still be displayed to the interview at the bottom line of the screen.

BEEPER Command

You also can turn the error beep off by using the BEEPER command in your source questionnaire program.

Example

```
.QUESTION = Q1, TYPE = NUMBER
...
.ANSWER = 5
.NEXT

.BEEPER = OFF

.QUESTION = Q2, TYPE = DATE
...
.ANSWER
.NEXT
```

Using the BEEPER command in your program gives you more flexibility because you can use it to turn the error beep off (or on) for an individual question, a series of questions, or all the questions in your questionnaire. If you use it within a question, it will change the setting for only that question. If you use it outside a question, it will change the settings for all of the following questions or until another BEEPER command is used outside a question.

In the last example, the error beep will be turned on for question Q1 because the default setting has not been changed. The error beep for question Q2 will be turned off because it was preceded by the BEEPER command. The error beep will also be turned off for any following questions since the BEEPER command was used outside of a particular question.

Option /F, Backup File Name

The COLLECT program always saves a record of an interview on the same disk (and in the same directory) and using the same file name as your compiled questionnaire. The extension is changed to DAT, to indicate that it is a data file. You also can tell the COLLECT program to automatically make a backup copy of the data file by entering a backup file name in the Startup Menu or by using the /F option when starting the program at the DOS command line. After the COLLECT program adds a record to the primary data file, it will add the same record to the end of the backup data file.

You should give the backup file a name that is different from the primary data file, and you may specify that it be stored on another disk drive or in another directory.

Example

```
C:\> COLLECT A:SAMPLE /FB:SAMPLBAK.DAT
```

In this example, the COLLECT program will use a questionnaire program called SAMPLE.QPL, which is on the disk in the A: drive. The primary data file will be called SAMPLE.DAT, and it will be stored on the disk in the A: drive. The /F op-

tion has been used to tell the COLLECT program to also create a backup data file. It will be called SAMPLBAK.DAT, and it will be stored on the disk in the B: drive.

You must specify the backup file name every time you start the COLLECT program in order to have all the records written to both the primary and backup data files. Records will be written to a backup file only when you tell the COLLECT program to do so.

BACKUP Command

You may instead tell the COLLECT program to create a backup data file by using the BACKUP command in your source questionnaire program. It may be used on any command line, but it is generally used near the beginning of a program.

Example

```
.BACKUP = B:SAMPLBAK.DAT

.QUESTION = Q1, TYPE = DATE
. . .
.ANSWER
.NEXT
```

In this example, the BACKUP command has been used to tell the COLLECT program to create a second copy of the data file, called SAMPLBAK.DAT and to write each record to the disk in drive B:. Since this command is in your source questionnaire program, the COLLECT program will always add a copy of each interview record to the end of the backup data file after it writes the record to the primary data file.

Option /H, Display Command Line Options

You can tell the COLLECT program to list all of the command line options on the screen if you start it with the /H option. Once the list is displayed, you can press any key to quit the COLLECT program and return to DOS. All the QPL programs (COMPILE, COLLECT, NCOLLECT, KEYPUNCH, MONITOR, and CONVERT) will display their command line options if you start them with the /H option.

Option /O#, Set OPTION Number

The /O option is used to set the value of the OPTION keyword. This keyword can be used by any QPL command that uses numeric expressions such as the IF, COMPUTE, or COMPUTEIF commands. The value of the OPTION keyword will only change if you restart the COLLECT program with a different /O value.

This capability is typically used to customize the skip path of a questionnaire from the DOS command line rather than by an interviewer entry. This allows you,

for example, to combine several versions of the same questionnaire (from the interviewers perspective) into one questionnaire program.¹

Option /P#, Enter Password
Number

The password option must be used to start a questionnaire in which a password has been specified.² If an interviewer does not enter a password number in this case, the COLLECT program will display an error message and then return to the DOS command line without starting the questionnaire.

The password must be a number from 1 to 32,767, and the number must be typed immediately following the option letter.

Example

```
C:\> COLLECT A:SAMPLE /P347
```

In this example, the COLLECT program will load a questionnaire called SAMPLE.QPL, which is on a disk in the A: drive, and which requires that password number 347 be given to start the program.

Option /Q, Do Not Display
Question Number

Normally, the COLLECT program will display the number and name of each question on the top-left corner of the screen. This is useful if the interviewer needs to make a note about a particular question or when you are developing a new questionnaire. The /Q option tells the COLLECT program not to display the question number and name on the interviewers screen.

Option /R, Do Not Restart
Interview

After the interviewer answers the last question in your questionnaire, the COLLECT program will add the interview record to the data file, and then display a dialogue box that asks if he or she would like to begin another interview. If the interviewer answers yes, the questionnaire will be restarted at the beginning, and all of the answers will be reset to their default values. If the interviewer answers no, the COLLECT program will end, and control will be returned to the DOS command line. You can use the /R option when starting the COLLECT program to suppress this dialogue box and return control to the DOS command line immediately after the interviewer answers the last question.

Option /T, Do Not Display
Program Title

Normally, the name of the COLLECT program, version number, and release date are briefly displayed on the screen in a blue box as the COLLECT program loads your questionnaire. Starting the COLLECT program with the /T option tells the COLLECT program not to display this title box as it loads your questionnaire.

¹See chapter 4, "OPTION Keyword."

²See chapter 8.

Option /U, Set Uppercase Control Off

Normally, the COLLECT program will let the interviewer enter only uppercase letters in STRING questions, regardless of whether the Shift or Caps Lock keys are pressed.³ You can let the interviewer to enter both upper and lowercase letters by checking “Use upper and lowercase letters” at the Startup Menu or by using the /U option when starting the COLLECT program at the DOS command line.

UPPERCASE Command

You may also change the uppercase setting with the UPPERCASE command in your source questionnaire program. Like the BEEPER command, you can use the UPPERCASE command to change the uppercase setting for all of your questions, a range of questions, or individual questions. If you use it within a question, it only affects the setting for that question. If you use it outside a question, it affects all of the following questions, or until another UPPERCASE command is used outside a question.

Example

```
.QUESTION = Q1, TYPE = STRING
.UPPERCASE = OFF
...
.ANSWER = 10
.NEXT

.QUESTION = Q2, TYPE = STRING
...
.ANSWER = Q1
.NEXT
```

In this example, the UPPERCASE command was within question Q1 to allow the interviewer to type both upper and lowercase letters. Since it was used within a question, the default setting will not be changed. Question Q2 will use the original default setting and only allow the interviewer to enter a response using uppercase letters.

Option /W, Set Workstation Name

This option is used to set the interviewer’s account name. This name can be automatically copied as a response to a STRING question (when used with the WORKSTATION subargument).⁴

This option provides compatibility for questionnaire programs you write for the network version of this program, NCOLLECT, which automatically uses the interviewer’s network account name. Here, it lets you manually set an account name for an interviewer who is using your questionnaire in a non-network environment.

³The UPPERCASE command has no affect on OPENEND questions.

⁴See chapter 3, “Short Answer.”

Screen Display

The COLLECT program displays one question at a time on the interviewer's screen. A question is displayed until the interviewer enters a response, then the COLLECT program will display the next question specified by the skip instructions. It automatically handles many of the tasks of displaying your questions on the screen:

- Question text is printed inside a box that is centered on the screen.
- The space where the interviewer enters the respondent's answer is highlighted with a bright background color.
- The question number and variable name are shown in the upper-left corner of the screen.
- The current time and date are shown in the upper-right corner of the screen.
- Error messages are displayed at the bottom of the screen with a bright background.

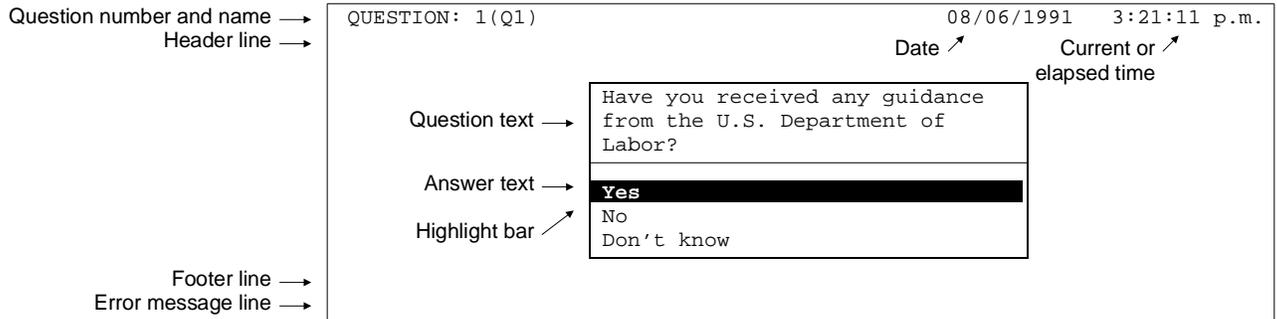
Question text is displayed in the box exactly as it is written in your source program, that is, the lines of your question will break exactly as you wrote them originally.

Example

```
Source questionnaire program

.QUESTION = Q1, TYPE = MULT
Have you received any guidance
from the U.S. Department of
Labor?
.ANSWER
Yes
No
Don't know
.NEXT
```

COLLECT Screen Display



There are several other aspects of the screen display that you can modify using commands in your source questionnaire program. You can

- change the colors that are used for computers that have color monitors,
- change the clock display to show the elapsed time while particular questions are shown on the screen, and
- add one or two lines on the screen that constantly display certain information (such as the name of the respondent).

Setting Screen Colors

The COLLECT program automatically determines whether the interviewer's computer has a color or monochrome graphic display card installed. If it is color, the COLLECT program will automatically use its default set of colors, or the colors you have specified, to display your questions on the screen. If it is a monochrome card (such as a Hercules card), questions will be displayed in black and white, regardless of any color commands. The names of the available colors are shown in the following table.

Screen Colors

BLACK	GRAY
BLUE	LIGHTBLUE
GREEN	LIGHTGREEN
CYAN	LIGHTCYAN
RED	LIGHTRED
MAGENTA	LIGHTMAGENTA
BROWN	YELLOW
WHITE	BRIGHTWHITE

Some computers, however, have color cards attached to monochrome monitors. In this case, the COLLECT program will display the questions in color, but they may be unreadable on the monochrome monitor screen. The COLLECT program can determine only what type of graphics card is installed in the interviewer's computer, not what type of monitor is attached to the card. In this case, the COLLECT program should be started with the /B option to force it to use black and white, or set the color commands to set the screen colors to black and white even if the interviewer is using a color monitor.

Four color commands are used to control the screen display: FOREGROUND, BACKGROUND, HIGHLIGHT, and BORDER. These commands may be used to set the colors for all of your questions, a range of questions, or individual questions. If a color command is used within a question, it will change the setting only for that question. If it is used outside a question, it will change the setting for all of the following questions, or until the command is used again outside of a question.

FOREGROUND Command

The color of the text of a question is set using the FOREGROUND command. The default is white for both color and monochrome monitors.

BACKGROUND Command

The color of the background of the question box is set using the BACKGROUND command. The default is blue for color monitors and black for monochrome monitors.

HIGHLIGHT Command

The color of the question number and name, clock, error messages, and information lines are set using the HIGHLIGHT command. The default color is cyan for color monitors, and white for monochrome monitors. The background is always black. These colors are reversed for error messages.

BORDER Command

The color of the lines of the question box are set using the BORDER command. The default is light blue for color monitors and white for monochrome monitors.

Example

```
.FOREGROUND = BLUE
.BACKGROUND = WHITE
.HIGHLIGHT = LIGHTGREEN
.BORDER = BLUE

.QUESTION = Q1, TYPE = NUMBER
. . .
.ANSWER = 8
.NEXT
```

```
.QUESTION = Q2, TYPE = NUMBER
.BORDER = RED
...
.ANSWER = Q1
.NEXT

.BACKGROUND = RED, FOREGROUND = WHITE

.QUESTION = Q3, TYPE = NUMBER
...
.ANSWER = Q1
.NEXT

.BACKGROUND = BLACK

.QUESTION = Q4, TYPE = NUMBER
...
.ANSWER = Q1
.NEXT
```

All of the default colors have been changed in this example with the color commands used at the beginning of the questionnaire. The COLLECT program will display question Q1 using blue text on a white background with a blue border. The question number and variable name will be displayed at the top of the screen in light green.

The border color for question Q2 will be changed to red. This color command affects only question Q2 since it was used inside the question.

The default foreground and background questions are changed after question Q2 to white and red. These settings will affect question Q3, but its border will still be blue according to the previous setting.

The background color is reset again after question Q3. Thus, question Q4 will be displayed in white text on a black background with a blue border. Any following questions also would be displayed in these colors.

CLOCK Command

The amount of time that a question has been displayed on the screen can be shown by the COLLECT program if the CLOCK command is set to "ON." You may use this command within a question to show how long it has been displayed on the screen. You also may use it to show the elapsed time for each of a range of questions by setting it to "ON" before the first question in the range, and setting to "OFF" after the last question in the range.

The elapsed time, shown in hours, minutes, and seconds, is displayed in the upper-right corner of the screen. The time always starts at zero when the question is displayed. If the interviewer backs up to a question, the time is reset to

zero. This time, however, is not recorded by the COLLECT program. It only is displayed for the convenience of the interviewer. Showing the elapsed time is useful, for example, if you want the interviewer to give up on a respondent who placed him or her on hold for more than three minutes.⁵

Example

Source Program	Questionnaire Format
<pre>.QUESTION=Q1, TYPE=MULT .CLOCK=ON Has the operator answered your call within three minutes? .ANSWER Yes, call answered No, not answered .NEXT</pre>	<pre>1. Has the operator answered your call within three minutes? (CHECK ONLY ONE ANSWER) __ 1. Yes, call answered __ 2. No, not answered</pre>

In this example, the interviewer already has been placed on hold. If the call is not answered within three minutes, the interviewer is instructed to answer “No, not answered.” The interviewer can tell that three minutes have elapsed because the CLOCK was used to display the time that has elapsed since the question was first displayed.

**HEADER and FOOTER
Commands**

One or two information lines can be displayed constantly during the interview using the HEADER and FOOTER commands. The HEADER command displays an information line at the top of the screen, and the FOOTER command prints an information line at the bottom of the screen. These commands may be used only once in your program. That is, an information line cannot be changed for individual questions.

The header and footer commands are programmed just as if they were question text lines, except that the text you want displayed must be delimited by double quotes. An information text line may contain only text, or it may include references to the answers of questions. The current answers to a question can be included in the information line by typing its variable name, delimited by square brackets, on the text line where you would like the answer to appear.

⁵If you want to measure how long an interviewer has been placed on hold, you should use TIME questions to record the starting and ending times. See chapter 3.

Example

```
.HEADER = "Respondent: [NAME], Organization: [COMPANY]"
.FOOTER = "Audit Company Survey"

.QUESTION = NAME, TYPE = STRING
Enter respondent's name.
.ANSWER = 20
.NEXT

.QUESTION = COMPANY, TYPE = STRING
Enter organization name.
.ANSWER = 30
.NEXT
```

In this example, a header line will be displayed at the top of the screen that shows the respondent's name and the name of his or her organization. The information on an individual respondent will be taken from questions NAME and COMPANY. Thus, this information will be blank until the interviewer answers these questions. For example, if the interviewer entered "Joe Auditor" and "ABC Auditing Corporation" to these questions, the header line would be displayed as: "Respondent: Joe Auditor, Organization: ABC Auditing Corporation."

A footer line has also been used in this example. Since it contains only text (*i.e.*, it does not refer to the answers of any questions), it will always be displayed at the bottom of the screen as "Audit Company Survey."

Organizing the Interviewer's Programs

Special care should be given to organizing the interviewer's disks. Typically, the interviewer's programs and data files are set up on floppy disks, making it easier to combine the data files and to protect the programs and data files from other users.

The only programs the interviewer needs are the COLLECT program and your compiled questionnaire. The sample setup below shows one way to organize the files. The COLLECT and questionnaire programs should be copied to a floppy disk that has been formatted with the disk operating system (*i.e.*, using the /S option with the DOS FORMAT command) if the interviewer's computer does not have a hard disk. If it does have a hard disk, copying the operating system is not necessary. The primary data file will also be saved on this disk. Backup data files should be written to a second disk that has been formatted without the operating system.

Sample Disk Setup

```
Disk A:                               Disk B:
-----                               -----
(Operating System)                   (backupfile.DAT)
COMMAND.COM
GO.BAT
COLLECT.EXE
filename.QPL
(filename.DAT)
```

Note: The data files will not be created until the COLLECT program is run for the first time.

Finally, you can write a batch program that starts the COLLECT program with the name of your questionnaire program and any particular options, such as writing a backup data file. In the sample batch program below, the interviewer needs only to type “GO” to start the COLLECT program. The DOS COPY command is used at the end of this batch file program to make copies of the open-ended question files.

Sample Start-Up Batch File (GO.BAT)

```
ECHO OFF
CLS
ECHO *****
ECHO ** Title of Survey **
ECHO *****
ECHO Please enter the current date if
ECHO it is not set correctly.
DATE
ECHO Please enter the current time if
ECHO it is not set correctly.
TIME
COLLECT TEST1.QPL /FB:TEST1BAK.DAT
COPY *.OTX B:
```

Correct Time and Date

The COLLECT program automatically checks to see if the correct date has been set on the computer’s internal clock. If the date is less than 1988, the COLLECT program will display an error message and then return control of the computer to the DOS command line without starting the questionnaire. You can also use a batch file to ask the interviewer to check the computer’s clock, as shown in the above example, before starting the COLLECT program.

Free Disk Space

The COLLECT program also checks to see that the disks have enough room to store the data files before it will proceed with an interview. If there is not enough room to write another record on a disk, the COLLECT program will display an error message on the screen and return to DOS.

You also should monitor the disks closely if your questionnaire program uses many OPENEND questions. Since the interviewer may enter a large amount of nar-

Quitting the COLLECT Program

rative in response to each of these questions, the COLLECT program cannot accurately predict when there will be too little free disk space available to store a new interview. You should check the interviewers' disks regularly to insure that they have enough room for additional interviews (especially if they are using 5¼-inch double-density disks, which hold less information than high-density or 3½-inch disks).

The COLLECT program does not allow the interviewer to quit an interview until all of the questions have been answered. After the interviewer has answered the last question, the COLLECT program will write the data file to the disk and then ask the interviewer if he or she would like to begin a new interview. If the interviewer says no, the COLLECT program will end and control will be returned to DOS. If he or she says yes, the questionnaire will be restarted at the beginning, and all of the answers will be reset to their default values.

An interviewer may exit the questionnaire early by pressing the Escape key. You can use the ESCAPE command to tell the COLLECT program to jump to a particular question when the Escape key is pressed. Generally, you should program it to jump to a question near the end of the questionnaire that asks the interviewer if he or she wishes to save the responses in the data file.⁶ If you do not use the ESCAPE command, the COLLECT program will skip to the last question in the questionnaire.

You also may use the ESCAPE keyword in a COMPUTE command to record the number of the question that was displayed on the screen when the interviewer pressed the Escape key. This is often done when using a control file to restart an incomplete interview at the point where it was interrupted.⁷

The skip pattern is not affected when the interviewer presses the Escape key. Any questions that were on the skip path will retain their default answers if the interview is saved in the disk file.

⁶See chapter 4.

⁷See chapter 6.

Running the NCOLLECT and MONITOR Programs

The NCOLLECT program is the local area network (LAN) version of the COLLECT program for computer-aided telephone interviewing (CATI). While it has same the functionality of the COLLECT program¹ and appears to the interviewer to be the same program, it takes advantage of a key network capability: sharing data files.

The NCOLLECT program is designed to allow two or more interviewers to share the same QPL questionnaire, data file, and control list file at the same time. This ability simplifies the setup and administration of telephone interviews since you do not need to distribute individual disks to each interviewer and then later merge their data files into one file.

The NCOLLECT program further helps you manage your project because it also keeps track of when interviewers are conducting an interview, who they have called, and whether the interview was completed. You can use the QPL MONITOR program to view this information from your workstation.² It automatically updates this information on your screen as interviews are started and completed.

You do not need to anything special in your questionnaire program to take advantage of the added functionality of the NCOLLECT program. You may, for example, let some of your interviewers run your questionnaire with the COLLECT program on stand-alone PCs, such as laptops, while others run it with the NCOLLECT program on networked PCs. Your questionnaire program and the data files that are created are identical.

Warning!

You should never attempt to access the data file that contains the interview records with the QPL KEYPUNCH or CONVERT programs or with your statistical analysis software while interviewers are using the system. These programs are not designed to share data files. Using them could prevent an interviewer from being able to save an interview record. You should always first copy the data file to another location, which is not shared with others, before editing or analyzing the data.

Novell NetWare

Although the NCOLLECT was designed to work with the Novell NetWare LAN operating system, you should be able to run it successfully with any PC-based LAN. The only Novell-specific feature it uses is Novell's user account name function, which you may disable using the /W or /Z command line options when start-

¹See chapter 9.

²See "Using the MONITOR Program," later in this chapter.

ing the NCOLLECT program (see below). The file sharing capabilities that NCOLLECT uses should be common to all PC LANs.

Starting the NCOLLECT Program

Unlike the COLLECT program, you must specify the name of your questionnaire on the DOS command line when you start the NCOLLECT program. Generally, you should create a DOS batch file or a Windows icon to make it easy for the interviewer to start your questionnaire with your particular command line options.

Example Batch File

```
ECHO OFF
CLS
F:\APPS\QPL\NCOLLECT F:\PROJECT\CATI\JOB1.QPL /D100
```

In this batch file example, the full path names to the NCOLLECT program and the questionnaire, JOB1.QPL, are specified. This is useful because it allows your batch file to run without depending on the current default directory setting or the DOS PATH environment variable. The /D option is also used to limit the number of outstanding control file cases that will be displayed to the interviewer to 100.

You could instead use the last line of this batch file as a Windows icon command line which would let your interviewers start your questionnaire from the Windows Program Manager.³ The following table lists all of the NCOLLECT command line options. They are discussed in greater detail later in this chapter.

DOS Command Line Options

```
NCOLLECT filename[.QPL] [/B] [/C] [/D#] [/E] [/H] [/O#]
           [/P#] [/Q] [/R] [/T] [/U] [/Wname] [/X#] [/Z]

/B        Set screen colors to black and white
/C        Use first available control file case
/D#       Maximum number control cases displayed (default 50)
/E        Set error beep off
/H        Display command line options
/O#       Set OPTION keyword number
/P#       Enter password number (range 1 to 32767)
/Q        Do not display question number line
/R        Do not restart interview
/T        Do not show program title
/U        Set uppercase control off
/Wname    Set workstation name
/X#       Set shared file access retry time (default 10 seconds)
/Z        Do not ask network for workstation name
```

Note: The brackets indicate optional information. The lowercase words indicate the type of information required. The “#” character indicates where a number is required.

³See “Creating and Deleting Program Items,” in the Windows Program Manager Help.

Files Created by NCOLLECT

The NCOLLECT program will create three or more files the first time you run it with a questionnaire. These files will be put on the same disk drive and in the same directory, but will have new file name extensions to indicate what type of file they are.

Files Created by NCOLLECT

<i>Extension</i>	<i>Type of File</i>
DAT	A fixed format data file containing interview records.
OTX	Files that hold responses to OPENEND questions. Each file holds the response to one question for one interview. The file name indicates the name of the question and the case number of the interview.
CAT	A catalog of information on who is currently using the questionnaire program. A file size of zero indicates that no one is using the questionnaire. A file size greater than zero indicates that one or more people are using the system. This file should never be deleted while the system is operation.
JRN	A journal of information on who has used the questionnaire. This file is not required to operate NCOLLECT and may be safely deleted at any time.
CNT	A text file that holds the last number used by the COUNT keyword. This file is only created if the COUNT keyword was used in the questionnaire program.

Generally, you do not need to be concerned about the creation of the data, catalog, journal, and count files. These are all designed to work automatically in a network environment without any special attention in your QPL program.⁴

Open-Ended Text Files

The OPENEND response files (OTX extensions) are created only if your questionnaire has OPENEND questions. Further, a file for a particular case is created only if the interviewer asks that OPENEND question. If the interviewer skips over an OPENEND question, the response file for that question will not be created.

OPENEND response files are also different from the other files in that they are not intended to be shared. That is, another interviewer should never be using the same open-ended response file at the same time as another interviewer. The NCOLLECT program manages this situation by giving each OTX file a unique name. The name is made unique by using both the question name (the first four characters of the name) and a case identification number (the last four digits of the name) to prevent duplicates.

⁴They do, however, need to be set up correctly when you run your survey with two or more interviewers on a network. See "Network Setup Issues," later in this chapter for more information.

In order for this scheme to work, you must make sure that each case is given a unique identification number.⁵

Tracking Interviews

The NCOLLECT program keeps track of which interviewer is working on which case by using catalog and journal files. These files are created automatically by the NCOLLECT program. They will have the same name as your questionnaire program, but will have CAT and JRN extensions to identify them.

Catalog File

The catalog file (CAT extension) is checked each time an interviewer starts a new interview. It holds the time that each interviewer started a new interview. This information is also used by NCOLLECT to prevent an interviewer from attempting more than one case at one time.⁶

The catalog file automatically gives you added control over what cases your interviewers may work on if your questionnaire uses a control file.⁷ Here, NCOLLECT will only let interviewers select cases that are (1) still outstanding, according to how you programmed the control file SHOWIF command, and (2) not currently being used by another interviewer.

When an interviewer finishes an interview, NCOLLECT removes this information from the catalog file and puts it into the journal file, along with information on when the interviewer finished and whether an interview record was saved.

Journal File

The journal file (JRN extension) lists every instance when an interviewer checked a case in and out. It lists the interviewer's workstation name, the control file case number (if used), and the starting and ending time for each interview. Although the journal file is not used to control which cases the interviewers may use, it is useful for monitoring what the interviewers are doing. The JRN file may be safely deleted at any time without affecting the operation of your project. You may want to delete it if it becomes very large in order to speed up the time it takes the NCOLLECT program to check in cases.

⁵See chapter 3, "Open-Ended Answer," for information about the OPENEND question, IDENTIFY command, and COUNT keyword.

⁶NCOLLECT can only do this when you are using interviewer workstation names (either by running the system on a Novell network or by setting each interviewer's workstation name at the command line using the /W option). If the workstation name is not set, one interviewer may start multiple copies of the questionnaire.

⁷See chapter 6, "Scheduling Interviews."

You may safely view the contents of the catalog and journal files using the MONITOR program (which is described later in this chapter) as interviewers are using the system to see who is currently working on interviews and what interviews have been attempted or completed. You should not attempt to view their contents using DOS EDIT or your word processor because they could damage the files as well as cause file sharing problems for interviewers who are currently working on cases.

Options

There are a number of options you may select when starting the NCOLLECT program that modify how it works. An option must always be written with a slash at the beginning, and all of the options and file names you use must be separated on the line by at least one space.

Option /B, Set Screen Colors to Black and White

The NCOLLECT program displays your questions on your display screen in color if your computer has a color graphics display card. It automatically switches to black and white if your computer has a monochrome display card, such as a Hercules graphics card. Some computers, however, have a color graphics card connected to a monochrome monitor (IBM/PC portable computers and many laptop computers are configured this way). In this instance, the NCOLLECT program would use colors, but they may be unreadable on the monochrome screen. You can force the NCOLLECT program to use black and white by adding the /B option to the end of the command line when starting the program, or checking “Black and white screen,” at the Startup Menu.

All of the QPL programs will use black and white if you start them with the /B option.

Option /C, Use First Available Control File Case

This option tells the NCOLLECT program to automatically use the first outstanding case in a control file instead of displaying a list of available cases to the interviewer. The SHOWIF command in the control questionnaire file determines when a case is available (that is, whether it is outstanding or completed). This option can simplify the operation of your project if it is not important which case an interviewer chooses.

Option /D, Limit Display of Control File Cases

By default, the NCOLLECT program will only display the first 50 outstanding cases in a control file. Limiting the number of cases speeds up the display of the control file list. You can use the /D option to change the default to another value. If you want NCOLLECT to always display all of your outstanding control file cases, use the /D option to set the maximum to the total number of cases in your

control file. You must type your new number immediately after the letter “D,” as shown in the example batch file above.

Option /E, Turn Error
Beep Off

The NCOLLECT program will beep and display a message on the bottom line of the monitor screen if the interviewer makes an error, such as trying to enter a letter into a numeric question or entering a value that is too high or too low. You can turn this beep off by using the /E option when starting the NCOLLECT at the DOS command line.⁸ Error messages, however, will still be displayed to the interview at the bottom line of the screen.

Option /H, Display
Command Line Options

You can tell the NCOLLECT program to list all of the command line options on the screen if you start it with the /H option. Once the list is displayed, you can press any key to quit the NCOLLECT program and return to DOS. All the QPL programs (COMPILE, COLLECT, NCOLLECT, KEYPUNCH, MONITOR, and CONVERT) will display their command line options if you start them with the /H option.

Option /O#, Set OPTION Number

The /O option is used to set the value of the OPTION keyword. This keyword can be used by any QPL command that uses numeric expressions such as the IF, COMPUTE, or COMPUTEIF commands. The value of the OPTION keyword will only change if you restart the NCOLLECT program with a different /O value.

This capability is typically used to customize the skip path of a questionnaire from the DOS command line rather than by an interviewer entry. This allows you, for example, to combine several versions of the same questionnaire (from the interviewers perspective) into one questionnaire program.⁹

Option /P#, Enter Password
Number

The password option must be used to start a questionnaire in which a password has been specified.¹⁰ If an interviewer does not enter a password number in this case, the NCOLLECT program will display an error message and then return to the DOS command line without starting the questionnaire.

The password must be a number from 1 to 32,767, and the number must be typed immediately following the option letter.

Option /Q, Do Not Display
Question Number

Normally, the NCOLLECT program will display the number and name of each question on the top-left corner of the screen. This is useful if the interviewer needs to make a note about a particular question or when you are developing a

⁸See also chapter 9, “BEEPER Command.”

⁹See chapter 4, “OPTION Keyword.”

¹⁰See chapter 8.

new questionnaire. The /Q option tells the NCOLLECT program not to display the question number and name on the interviewers screen.

Option /R, Do Not Restart Interview

After the interviewer answers the last question in your questionnaire, the NCOLLECT program will add the interview record to the data file, and then display a dialogue box that asks if he or she would like to begin another interview. If the interviewer answers yes, the questionnaire will be restarted at the beginning, and all of the answers will be reset to their default values. If the interviewer answers no, the NCOLLECT program will end, and control will be returned to the DOS command line. You can use the /R option when starting the NCOLLECT program to suppress this dialogue box and return control to the DOS command line immediately after the interviewer answers the last question.

Option /T, Do Not Display Program Title

Normally, the name of the NCOLLECT program, version number, and release date are briefly displayed on the screen in a blue box as the NCOLLECT program loads your questionnaire. Starting the NCOLLECT program with the /T option tells it not to display this title box as it loads your questionnaire.

Option /U, Set Uppercase Control Off

Normally, the NCOLLECT program will let the interviewer enter only uppercase letters in STRING questions, regardless of whether the Shift or Caps Lock keys are pressed.¹¹ You can let the interviewer enter both upper and lowercase letters by using the /U option when starting the NCOLLECT program at the DOS command line.¹²

Option /W, Set Workstation Name

If your network is running under Novell NetWare (or other compatible network software), the NCOLLECT program will automatically set the workstation name to the interviewer's account name.¹³ This name can be used as a response to a STRING question (when used with the WORKSTATION subargument).¹⁴ This allows you to reliably capture the name of the interviewer in each interview record.

You can use the /W option, however, to replace the Novell account name with a name of your own choosing. The /W option both sets the workstation name and tells NCOLLECT not to ask the LAN operating system for the user's account name. Your new name must be one word and may be up to 45 characters long. You must write the name on the DOS command line immediately after the "W."

¹¹The UPPERCASE command has no affect on OPENEND questions.

¹²See also chapter 9, "UPPERCASE Command."

¹³The account name is based on the first server to which he or she was connected.

¹⁴See chapter 3, "Short Answer."

Example

```
NCOLLECT JOB1.QPL /WSally
```

In this example, the /W command line option was used to set the workstation name to “Sally,” instead of the Novell account name.

Option /X, Set Retry Time

The NCOLLECT program will wait to get access to a file that is currently being used by another user for up to 10 seconds.¹⁵ This file may be either a data, catalog, or journal file. If this time expires before the program could access the file, the user is presented with a dialog box that asks if they want to try to access the file again or give up. If the user chooses to attempt to access a file again, the NCOLLECT program will wait for access for another 10 seconds. You may use the /X option to adjust this waiting time from 2 to 60 seconds.

Option /Z, Do Not Get Workstation Name

The /Z option tells the NCOLLECT program not to ask the network operating system for the user’s (i.e., interviewer’s) account name. You may need to use this option to prevent your computer from hanging if your network is not running under Novell NetWare. This may be necessary even if you do not plan to use the account name in your questionnaire program.

If your LAN uses other operating software, you should test the NCOLLECT program to make sure it can correctly retrieve a user account name. If it is unable to do so you should retry running the NCOLLECT program with the /W or the /Z command line options.¹⁶ Both of these options prevent the NCOLLECT program from making a system interrupt call which should allow the program to run on any network.

You may also use this option to set up your application so that it does not use workstation names to keep track of who has cases checked out. Starting the NCOLLECT program with the /Z option tells it create a unique user number each time an interviewer starts a questionnaire. If a number is already in use, it simply creates another number. Thus, this allows one user to run several versions of the questionnaire at the same time or start a new questionnaire even if they did not log out correctly on a previous questionnaire. While doing this prevents a user from ever getting an “ID in use” message, it prevents you from using the MONITOR program to identify who is using your questionnaire system.

¹⁵The NCOLLECT program ignores any keys the user may press while waiting to get access to a file.

¹⁶The NCOLLECT program actually first tests for the presence of the Novell operating system. If it is present, according to this test, it will ask for the user account name. These system interrupt calls cannot harm your network. The worst that can happen is that your network does not respond, making your PC appear to be locked up because it is waiting for a response to its system interrupt call which is not forthcoming.

Network Setup Issues

After you have finished developing your questionnaire, you will need to work with your local LAN administrator to install the NCOLLECT program and your questionnaire on your network. The LAN administrator has the knowledge and supervisory rights necessary to

- establish a shared, or public, directories on your file server,
- set directory rights and program and data file attributes,
- give you and your interviewers access to these directories, and
- determine the best way to start the program.¹⁷

The following discussion describes how this software is typically installed on a Novell network; your LAN administrator may need to modify this example to fit how your network is organized.

Create a Program Directory and Give Interviewers Rights

Put the executable QPL programs (minimally, the NCOLLECT.EXE and MONITOR.EXE programs) in public directory, such as F:\APPS\QPL. The following table lists the rights interviewers should be given to the program directory.

Program Directory Rights

Read
File scan

The attributes of the files in this directory should also be set to prevent someone with supervisory rights from accidentally deleting or renaming the program files and to indicate that they may be shared with multiple users.

Program File Attributes

Read only
Shareable

Create a Data Directory

Put the compiled questionnaire program (which has a QPL file name extension) in a shared data directory, such as F:\PROJECT\CATI. If questionnaire is uses a control file, also copy the control compiled questionnaire (a second QPL file) and its data file (which has a DAT file name extension) to this directory. The following table lists the rights interviewers should be given to the data directory.

¹⁷Such as from a locally maintained LAN menu, a batch file, or a Windows icon.

Data Directory Rights

Read
Write
Create
Modify
File scan

The attributes of the files in the data directory also should be set to prevent interviewers from accidentally deleting or corrupting them. The following table shows how each of the files should be flagged. (Note: The DAT, CNT, CAT, and JRN files are not created until the NCOLLECT program is run with the questionnaire for the first time. You may wish to run it, saving a dummy case to create a DAT file in order to set the file attributes before interviewers begin using the system.)

Data File Attributes

<i>Extension</i>	<i>Attributes</i>
QPL	Read only ¹⁸ Sharable
DAT, CNT, CAT, JRN	Read write Sharable Delete inhibit Rename inhibit

Note: OTX files are not shared and do not need to be specially flagged.

Give Users Access

The final installation step is to give the interviewers an easy way to start the NCOLLECT program with the questionnaire. See “Starting the NCOLLECT Program,” at the beginning of this chapter.

Trouble Shooting

Running a CATI in a network environment solves many administrative problems, but also requires more careful attention to how the system is setup. Since files are being shared, it is important that you follow the guidance above in order to prevent accidental damage to your questionnaire or data files. Even with these precautions, problems can sometimes arise. The following discussion lists other problems that may occur and their remedies.

Computer Locks Up

This problem may occur if you are running NCOLLECT on a non-Novell network which does not support Novell’s system interrupt call for a user name. To fix this

¹⁸When you flag a file Read Only, NetWare automatically assigns the Delete Inhibit and Rename Inhibit attributes.

Cannot Create, Open, Edit, or Resize Files

problem, start the NCOLLECT program with the /W or /Z options to tell NCOLLECT not to check for the name. The /W command may also be used to manually set a interviewer name from the DOS command line. See the discussion of the /W and /Z command line options above.

As discussed in the previous section on setting file access rights, the NCOLLECT program needs to access five files in order operate:

- Questionnaire program,
- Data,
- Catalog,
- Journal, and
- Count files.

If the questionnaire is using a control file that lists the survey respondents, then the NCOLLECT program will also need to access two more files:

- Control questionnaire program, and
- Control data file.

The questionnaire files are only opened in a read-only mode while the other files may be opened in read-only or read-write modes. Basic file operation errors fall into four categories:

- Cannot create a file,
- Cannot open a file,
- Cannot edit a file, and
- Cannot resize a file.

File creation errors are usually due to a user not having rights create a file in the project directory. This may be fixed by setting the user rights as described in the previous section. Other errors that may occur when performing these file operations usually are caused by one of the two following situations.

Permission denied. This message indicates that another user has opened the same file. Since the NCOLLECT program never keeps a file open for more than about one second, getting this message repeatedly may indicate that another user has opened the file with a non-file sharing program, such as KEYPUNCH or CONVERT, that can keep the file open longer, preventing others from accessing it. Fix this problem by first retrying to access the file (using the dialog box that is displayed) and then finding the other user who has the file open and ask them to close it.

You may adjust the amount of time that the NCOLLECT program will use to access a file from 2 to 60 seconds by using /X command line option (see description above). By default, the NCOLLECT program will repeatedly try to access a file for up to 10 seconds before displaying this error message.

Generally, the data files in your project data directory should only be accessed with the NCOLLECT or MONITOR programs which are designed to share files. If you want to edit or analyze the data files, you should first copy them to another non-shared directory before accessing them. Further, you should only copy the files when no one is using the system. You can quickly tell when no one is conducting an interview because the catalog file will have a file size of zero bytes. You may also check the system status using the MONITOR program.

Bad file number. This message probably indicates that a file, which NCOLLECT is trying to update, has been flagged with a Read-Only attribute. Fix this problem by setting the file attributes to Shared and Read-Write as described in the previous section.

Cannot Check a Case In or Out

There are several error messages that may be displayed to an interviewer if there is a problem related to checking a case in or out.

Your workstation ID is already in use. This message indicates one of two things: (1) that another interviewer has the same workstation name and is currently conducting an interview, or (2) this interviewer did not correctly exit the system the last time he or she conducted an interview. The first cause should be relatively easy to check.

If an interviewer does not exit a questionnaire properly, they will be listed in the catalog file as still working on an interview. Exiting properly both saves an interview record and checks in the case. To diagnose this situation, you should use the MONITOR program (discussed below) to see what cases are currently being worked on. If you find, for example, that he or she is listed as still working on a case that he or she started yesterday, then it is clear that they did not exit properly (possibly by pressing the computer's reset or on/off buttons).

You may fix this situation by erasing the catalog file after you make sure no one else is currently conducting an interview. The catalog file will be automatically recreated the next time the NCOLLECT program is run.¹⁹

¹⁹ Instead of erasing the file, you may use the SQUASH.EXE utility program included on the QPL Master Program Disk to reduce the catalog file size to zero bytes. This has the same effect as erasing the file but preserves its additional network file attributes (i.e., Shareable and Read-Write).

Alternatively, you may set up your application so that it does not use workstation names to keep track of who has cases checked out. Starting the NCOLLECT program with the /Z option tells it create a unique user number each time an interviewer starts a questionnaire. If a number is already in use, it simply creates another number. Thus, this allows one user to run several versions of the questionnaire at the same time or start a new questionnaire even if they did not log out correctly on a previous questionnaire. While doing this prevents a user from ever getting an “ID in use” message, it prevents you from using the MONITOR program to identify who is using your questionnaire system.

Your case has not been checked out. This error message is caused when there is no record in the catalog file that the interviewer’s case was checked out. This error message should only occur if the catalog file was deleted while the interviewer was working on a case. This is a non-recoverable error. You should never delete the catalog file unless you know that no one is using the system.

This case is already checked out. This error message is displayed when an interviewer was beaten to a case by another interviewer. This may occur when your questionnaire uses a control file of respondents. Normally, the control list only displays cases that are still outstanding and not currently being used. If two interviewers are viewing the list at the same time, and select the same case, the interviewer who made their selection first will get the case and the second one will get this error message. To recover, the second interviewer only needs to restart the control list and make another selection.

Using the MONITOR Program

The MONITOR program is an interactive tool that you can use to see how your survey is progressing. It shows what cases your interviewers are working on and what cases they have completed. The information on the MONITOR screen is constantly updated so you can see when interviewers begin new cases or print a snapshot of the system status.

Sample MONITOR Screen Display

```

N:\PROJECTS\CATI\A13.QPL
Monitor Print Quit
ID Header Cntr Data Day Date Start Stop Len
No interviews are in progress.
vogelsang Mr. K.C. Koogler, New 98 127 Thu 09-14 08:48 08:50 140
vogelsang Mr. K.C. Koogler, New 98 127 Wed 09-13 13:04 13:21 1000
stephens Ms. Marianne C. Semen 117 124 Wed 09-13 09:42 09:48 358
warholic Mr. K.C. Koogler, New 98 127 Tue 09-12 12:01 12:03 84
warholic Mr. Charles Sindelar, 3 96 Mon 09-11 14:27 15:34 4017
warholic Harry Stackhouse, Ver 56 128 Thu 09-07 16:05 16:27 1322
warholic Harry Stackhouse, Ver 56 Thu 09-07 15:34 15:34 37
vogelsang Mr. K.C. Koogler, New 98 127 Thu 09-07 13:34 13:50 975
warholic Mr. Charles Sindelar, 3 96 Tue 09-05 11:29 11:42 812
warholic Howard B. Rosen, Gene 74 125 Tue 08-29 15:12 15:19 413
warholic Howard B. Rosen, Gene 74 125 Tue 08-29 15:11 15:12 33
warholic Howard B. Rosen, Gene 89 126 Tue 08-29 13:03 13:19 956
vogelsang Mr. Charles Sindelar, 3 Mon 08-28 14:57 14:57 27
warholic Howard B. Rosen, Gene 74 125 Thu 08-24 12:02 12:53 3061
warholic Howard B. Rosen, Gene 74 Thu 08-24 11:58 12:02 250
stephens Ms. LC Charlesworth, 33 119 Wed 08-23 10:48 11:09 1273
stephens Ms. Marianne C. Semen 117 124 Mon 08-21 13:35 14:15 2414
stephens C. Eric Hunter, Advan 48 123 Fri 08-18 09:08 09:40 1939
warholic Mr. Charles Sindelar, 3 96 Thu 08-17 09:27 09:34 422
Logged in: 0 Logged out: 230 Data records: 128 Delay: 22

```

The first line of the MONITOR display shows the name and location of the questionnaire, in this case called A13.QPL.

The next line is the command menu. Move the highlight bar using the left and right arrow keys and then select the highlighted option by pressing the Enter key.

The lines in the middle of the screen show the current status of your survey. Each line summarizes the status of each attempt an interviewer made to complete an interview. The most recent attempts are listed first. Attempts shown in bold at the top of the list indicate cases that are currently being conducted by an interviewer (these cases also do not show a “Stop” time since the interviewer has not finished yet). In the example above, no interviews are currently being conducted so MONITOR displays the message, “No interviews are in progress.”

Scroll the list up and down using the up and down arrow keys, and the Page Up and Page Down keys.

The last line displays several summary pieces of information. The “Logged in,” number indicates how many interviewers are currently conducting interviews. The “Logged out” number indicates how many interviews have been attempted (that is, how many times interviewers have started the NCOLLECT program whether or not they were able to successfully complete an interview). The “Data records” number indicates how many interview records have been stored in the main data file. Finally, the “Delay” number shows the number of seconds before the screen will be updated.

The following table summarizes what information is in each of the columns in the list.

Information Displayed by MONITOR

ID	Network account name of the interviewer (or name set with the /W command line option).
Header	Copy of the HEADER command line displayed on the interviewer's screen (usually used to display respondent information).
Cntr	Record number of control data file case (blank if a control file is not being used).
Data	Record number of interview in main data file (blank if no interview record was saved).
Day	Day-of-the-week interview was started.
Date	Month and day interview was started.
Start	Time interview was started (using a 24-hour clock).
Stop	Time interview was completed (blank if interview is not finished).
Len	Number of seconds to complete interview (blank if interview is not finished).

The date, start, and stop time information is based on an individual interviewer's computer clock. You should make sure that the date and time are correctly set on each interviewer's computer so that these times are accurately recorded.

Starting the MONITOR Program

You must specify the name of your questionnaire file when you start the MONITOR program. To make it easier to start the MONITOR program, you should create a batch file or Windows Program Manager icon that starts it with your particular command line options.

Example Batch File

```
ECHO OFF
CLS
F:\APPS\QPL\MONITOR F:\PROJECT\CATI\A13.QPL /D150
```

In this batch file example, the full path names to the MONITOR program and the questionnaire, A13.QPL, are specified. This is useful because it allows your batch file to run without depending upon the current default directory setting on the DOS PATH environment variable. The /D option is used to increase the number of cases that will be displayed from 100 (the default) to a maximum of 150 cases.

You could instead use the last line of this batch file as a Windows icon command line which would let your interviewers start your questionnaire from the Windows Program Manager.²⁰

The following table lists all of the MONITOR command line options.

DOS Command Line Options

```
MONITOR filename[.QPL] [/B] [/D#] [/H] [/T]

/B      Set screen colors to black and white
/D#     Maximum number cases displayed (default 100)
/H      Display command line options
/T      Do not show program title
```

Options

There are a number of options you may select when starting the MONITOR program that modify how it works. An option must always be written with a slash at the beginning, and all of the options and file names you use must be separated on the line by at least one space.

Option /B, Set Screen Colors to Black and White

The screen colors the MONITOR program uses are based on the color settings from the first question in your questionnaire.²¹ It automatically switches to black and white if your computer has a monochrome display card, such as a Hercules graphics card. Some computers, however, have a color graphics card connected to a monochrome monitor (many laptop computers are configured this way). In this instance, the MONITOR program would use colors, but they may be unreadable on the monochrome screen. You can force the MONITOR program to use black and white by adding the /B option to the end of the command line when starting the program, or checking “Black and white screen,” at the Startup Menu.

All of the QPL programs will use black and white if you start them with the /B option.

Option /D, Set Maximum Number of Cases to Display

The MONITOR program always displays the most recent interview attempts at the top of the list, up to a total of 100 attempts. The /D option is used to increase or

²⁰See “Creating and Deleting Program Items,” in the Windows Program Manager Help.

²¹See FOREGROUND, BACKGROUND, BORDER, and HIGHLIGHT commands.

decrease the number of cases that will be displayed. (It takes longer to display more cases.) This option does not affect the Print command. It will always print all of the attempts.

Option /H, Display
Command Line Options

You can tell the MONITOR program to list all of the command line options on the screen if you start it with the /H option. Once the list is displayed, you can press any key to quit the MONITOR program and return to DOS. All the QPL programs (COMPILE, COLLECT, NCOLLECT, KEYPUNCH, MONITOR, and CONVERT) will display their command line options if you start them with the /H option.

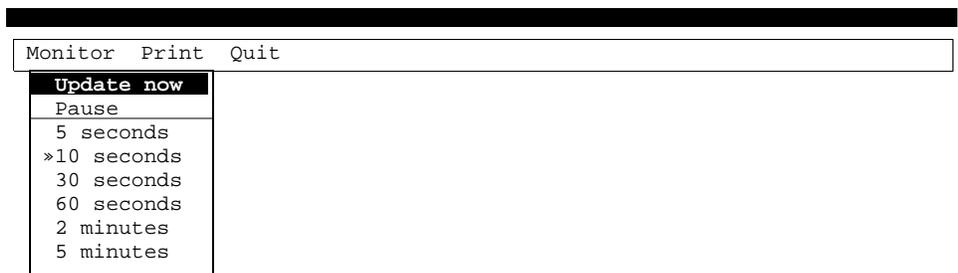
Option /T, Do Not Display
Program Title

Normally, the name of the MONITOR program, version number, and release date are briefly displayed on the screen in a blue box as the MONITOR program loads your questionnaire. Starting the MONITOR program with the /T option tells it not to display this title box as it loads your questionnaire.

Monitor Menu

The Monitor menu lets you set how frequently the MONITOR program will update the display. By default, the display will be updated every 10 seconds. Using this menu, you may change the delay from 5 seconds to 5 minutes. Move the highlight bar to the delay time of your choice and then press the Enter key select that time. The check-mark indicates the currently selected time. The elapsed time (in seconds) until the screen is updated is constantly displayed on the lower right-hand side of the screen.

Monitor Menu



Press the Esc (escape) key to close the Monitor menu.

Update Now

The "Update now" option tells the MONITOR program to immediately update the screen and reset the delay counter.

Pause

The "Pause" option tells the MONITOR program to stop updating the screen. If you select this option, a check-mark will be displayed next to this item on the menu and the word "Paused," will be displayed on the lower right-hand side of

the screen. The MONITOR program will remain in this paused mode until you select the Pause option again.

Print Menu

The Print menu lets you print the information on the MONITOR screen. Unlike the screen display, however, the printing options will print information about every attempted interview. Using the Print menu options, you may select to print a report that looks like the MONITOR screen or to export the data to a file in comma delimited format (which is suitable for importing into a spreadsheet or statistical program).

Print Menu

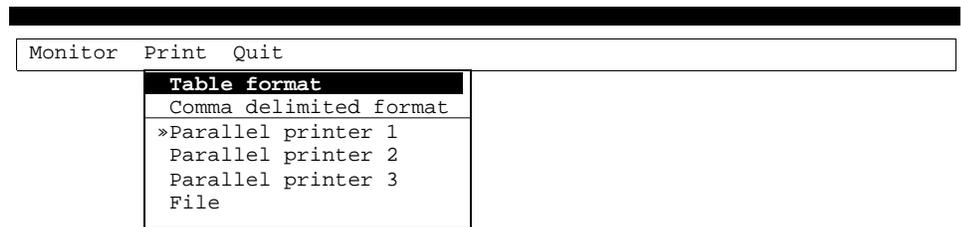


Table Format

The “Table format” option will print a report that looks just like the screen. The TITLE and SUBTITLE you set in your questionnaire program will be used to title the report. It will also be broken into numbered pages. It will be printed to the destination indicated by the check-mark in the lower half of the menu. By default, it will be printed to the printer that is connected to your computer’s LPT1: printer port. If you want to print to a different location, you should select the port, or file, before selecting the Table Format option.

Comma Delimited Format

The “Comma delimited format” option can be used to export the information on the MONITOR screen in a format that can be readily imported into a spreadsheet or statistical program for further analysis. You should change the output destination to the File option before selecting this option.

Output Destinations

The options in the lower half of the Print menu let you tell the MONITOR program where to print your report (according to how printers are connected to your computer via the LPT ports) or to copy the report to an ASCII text file. You should select one of these output destinations before selecting a report option.

To select an output destination, first highlight it and then press the Enter key. The check-mark indicates the current output destination.

If you select the File option, a dialog box will be displayed (after you select one of the two report options) that will ask for a file name for the report. It will suggest a name based on your questionnaire name, but will change the file name ex-

tension to MTX to indicate that the file is a MONITOR report text file. You may change this name if you wish.

Quit

Select this command to exit the MONITOR program.

Running the KEYPUNCH Program

The KEYPUNCH program allows you to

- enter data into a new record,
- retrieve and edit an existing record,
- delete a record,
- sort the data file, and
- print one or more of the records in an easy-to-read format.

The KEYPUNCH program has the same editing features as the COLLECT program, which make it easy to accurately enter and edit survey data. Instructions on operating the KEYPUNCH program are given in the interviewer's manual, *QPL Data Editing Program—Version 4.0*. Instructions are given here on the various options that may be used when starting the program, and a suggested method of organizing the programs and data files on the interviewer's disks.

There are two ways to tell the KEYPUNCH program which questionnaire you want to use. One way is to use the “Startup Menu,” which asks you for the name of your questionnaire and whether you need to reset the screen colors to black and white.

Startup Menu

The Startup Menu is displayed if you start the KEYPUNCH program without any options — that is, when you start it by typing “KEYPUNCH” and then pressing the Enter key.

Example

```
C:\> KEYPUNCH
```

The Startup Menu lets you edit the file name and option before you start the program. A highlight bar indicates which item in the menu you are pointing at. You can move the highlight bar up and down using the arrow keys on the right side of your keyboard or by pressing the Enter key.

The highlight bar always points to the source file name when you first start the program. You must type the name of your compiled questionnaire file in this space. Typing the QPL extension, however, is optional. The KEYPUNCH program will always load the file with this extension, even if you enter a file name with another extension. Use the Insert, Delete, and backspace keys to edit the source file name.

Press the spacebar to toggle the option on or off. A check mark indicates that the option is on.

You must press the Enter key to jump from the option to the “Begin” box. Press Enter at this point to start the program, or use the arrow keys to highlight “Cancel” to quit the program. Pressing the up arrow key will take you back to the file names and options. You also may press the Escape key at any time to quit the Startup Menu and return to DOS.

You may instead start the KEYPUNCH program with your questionnaire if you specify its file name at the DOS command line.

DOS Command Line Options

```
KEYPUNCH [filename[.QPL]] [/B] [/F] [/H] [/Ifilename.INI]
          [/L#] [/O#] [/P#]

/B        Set screen colors to black and white
/F        Print reports to a file
/H        Display command line options
/Ifile    Use printer initialization file (default is QPLPRINT.INI)
/L#       Parallel printer port (default is 1)
/O#       Set OPTION keyword value
/P#       Enter password number (range 1 to 32767)
/Wname    Set workstation name
```

Note: The brackets indicate optional information. The # character indicates where numbers are required. The lowercase words indicate what type of information is required.

Example

```
C:\> KEYPUNCH A:SAMPLE
```

In this example, the KEYPUNCH program will use the questionnaire program, SAMPLE.QPL, which is located on the disk in the A: drive.

Options

As noted above, there are a number of options you may select when starting the KEYPUNCH program. If you use an option at the command line, you must always precede the option letter with a slash. If an option requires a number, then the number must be typed immediately after the option letter. Finally, each option and its argument must be separated from other options and the file name on the command line by at least one space.

Option /B, Set Screen Colors to Black and White

The KEYPUNCH program displays information on your display screen in color if your computer has a color graphics display card. It automatically switches to black and white if your computer has a monochrome display card, such as a Hercules graphics card. Some computers, however, have a color graphics card connected to a monochrome monitor (many laptop computers are configured this way). In this instance, the KEYPUNCH program would use colors, but they may be unreadable on the monochrome screen. You can force the KEYPUNCH program to use black and white by adding the /B option to the end of the command line when starting the program, or checking "Black and white screen" on the Startup Menu.

All of the QPL programs will use a black and white screen if you start them with the /B option.

Option /F, Print to File

Normally, the KEYPUNCH program will print reports (the responses to one or more questionnaires) on the printer that is connected to your first printer port (i.e., LPT1:). You may use this option, however, to tell the KEYPUNCH program to print a report to a file. When you use one of the Print Menu functions, a dialog box will be displayed that will ask for a file name. It will suggest a name that is the same as your questionnaire, but with a new extension, KTX, to indicate that it is a KEYPUNCH text file. You may later print this file using the DOS PRINT command.

You may also select this option after you have started the KEYPUNCH program by select the "Print to File," option on the Print Menu.

Option /H, Display Command Line Options

You can tell the KEYPUNCH program to list all of the command line options on the screen if you start it with the /H option. Once the list is displayed, you can press any key to quit the KEYPUNCH program and return to DOS.

All the QPL programs will display their command line options if you start them with the /H option.

Option /I, Printer Initialization File

Printer initialization files are used to program your printer so that it will use a smaller type size and line width. This allows you to print more of your formatted

record listing on each page. By default, the KEYPUNCH program will load the initialization file called QPLPRINT.INI if it is located on the same disk and in the same directory as your questionnaire program. You may instead use the /I option to tell the KEYPUNCH program to load another file, that may be located on another disk or in another directory.¹

Option /L#, Set Printer Port

Normally, the KEYPUNCH will print reports on the printer that is connected to your first printer port (i.e., LPT1:). You may use this option to use another port.

Example

```
C:\> KEYPUNCH A:SAMPLE /L2
```

In this example, the /L option was used to tell the KEYPUNCH program to use the printer connected on LPT2: as the default printer.

You may also use the Print Menu options to change the printer port after you have started the KEYPUNCH program.

Option /O#, Set OPTION Number

The /O option is used to set the value of the OPTION keyword. This keyword can be used by any QPL command that uses numeric expressions such as the IF, COMPUTE, or COMPUTEIF commands. The value of the OPTION keyword will only change if you restart the COLLECT program with a different /O value.

This capability is typically used to customize the skip path of a questionnaire from the DOS command line rather than by an interviewer entry. This allows you, for example, to combine several versions of the same questionnaire (from the interviewers perspective) into one questionnaire program.²

Option /P#, Enter Password Number

The password option must be used to start a questionnaire in which a password has been specified.³ If an interviewer does not enter a password number in this case, the KEYPUNCH program will display an error message and then return to the DOS command line without starting the questionnaire.

The password may be a number from 1 to 32,767, and the number must be typed immediately following the option letter.

Example

```
C:\> KEYPUNCH A:SAMPLE /P347
```

¹See *QPL Data Editing Manual—Version 4.0* for more information on using printer initialization files.

²See chapter 4, “OPTION Keyword.”

³See chapter 8.

In this example, the KEYPUNCH program will load a questionnaire called SAMPLE.QPL, which is on a disk in the A: drive, and which requires that password number 347 be given to start the program.

Option /W, Set Workstation Name

This option is used to set the interviewer's account name. This name can be automatically copied as a response to a STRING question (when used with the WORKSTATION subargument).⁴

This option provides compatibility for questionnaire programs you write for the network version of this program, NCOLLECT, which automatically uses the interviewer's network account name. Here, it lets you manually set an account name for an interviewer who is using your questionnaire in a non-network environment.

Screen Colors

If the KEYPUNCH program is being used on a computer that has a color monitor, it will automatically use colored text and lines in the screen display. You can change the default colors that it uses by specifying other colors with the FOREGROUND, BACKGROUND, HIGHLIGHT, and BORDER commands in your source questionnaire program. Unlike the COLLECT program, which can change screen colors for each question, the colors for the KEYPUNCH screen will be based on the colors that you set for the first question in your program.⁵ If you display the full text of a question, however, by pressing the F2 key while editing, the text will be displayed using the colors you programmed.

Titles and Subtitles

It is generally useful to label the formatted record printouts with titles and subtitles. One title and one subtitle can be printed at the top of each record, and additional subtitles can be used to label groups of questions in the record, making it easier to review. These titles and subtitles are also used by the CONVERT program when it creates a formatted version of your questionnaire program and when it generates SPSS and SAS programs.

TITLE Command

A title line must be specified in your source questionnaire program using the TITLE command. It cannot be set by the interviewer while using the KEYPUNCH

⁴See chapter 3, "Short Answer."

⁵See chapter 9, for examples of how the color commands are used.

program. Only one title may be used in your program, and it must be delimited by double quotes. Finally, a title may not be longer than 70 characters.

SUBTITLE Command

Subtitles also must be specified in your source questionnaire program using the SUBTITLE command. Two kinds of subtitles may be used. One subtitle may be specified that will always be printed after the title line. This subtitle line cannot be more than 64 characters long. Subtitles can also be used to label groups of questions.

Example

```
.TITLE = "Summer Youth Employment and Training Survey"  
.SUBTITLE = "December 1987"  
  
.QUESTION = Q1, TYPE = DATE  
.SUBTITLE = "I. Background questions"  
  ...  
  ...  
  ...  
.ANSWER  
.NEXT
```

This example shows how the TITLE and SUBTITLE commands are used. Here, the title “Summer Youth Employment and Training Survey” will always be used to label the KEYPUNCH program’s printed interview records. The subtitle “December 1987” will always be printed immediately underneath the title on each page.

A second SUBTITLE command has been used to link a label to a particular question. Since the subtitle “I. Background Questions” is used within a question (*i.e.*, between a QUESTION command and its NEXT command), this subtitle will always be printed immediately above the question in the list of questions on the interview record. Thus, in order to label groups of questions with a subtitle, you need only to link the subtitle to the first question in the group. Only one subtitle can be used within a question.

Organizing the Interviewer’s Programs

Special care should be given to organizing the interviewer’s disks. Typically, the interviewer’s programs and data files are set up on a floppy disk, making it easier to combine the data files and to protect the programs and data files from other users.

The only programs the interviewer needs are the KEYPUNCH program and your compiled questionnaire. The sample setup below shows one way to organize the files. The KEYPUNCH and questionnaire programs should be copied to a floppy

disk that has been formatted with the disk operating system (*i.e.*, using the /S option with the DOS FORMAT command) if the interviewer's computer does not have a hard disk. If it does have a hard disk, copying the operating system is not necessary. The primary data file will also be saved on this disk.

Backup Data File

Unlike the COLLECT program, the KEYPUNCH program does not automatically backup interview records, even if the BACKUP command has been used in your source questionnaire document. There is a reason for this. Since the KEYPUNCH program can be used to change interview answers (and with that the possibility of making mistakes) you generally would not want it to automatically make the changes in your back up file and thus lose all of the original information. You should establish a procedure with your interviewer so that back up files are made under new names and at regular intervals.

Sample Disk Setup

```
Disk A:
-----
(Operating System)
COMMAND.COM
GO.BAT
KEYPUNCH.EXE
filename.QPL
(filename.DAT)
```

Note: The data file will not be created until the KEYPUNCH program is run for the first time (unless data were obtained using the COLLECT program).

Finally, you can write a batch program that starts the KEYPUNCH program with the name of your questionnaire program and any particular options, such as those needed to use a serial printer. In the sample batch program below, the interviewer needs only to type GO to start the KEYPUNCH program.

Sample GO.BAT Batch File

```
ECHO OFF
CLS
ECHO *****
ECHO ** Title of Survey **
ECHO *****
ECHO Please enter the current date and
ECHO if it is not set correctly.
DATE
TIME
REM Make back up copy of old data and OTX files.
IF EXIST TEST1.DAT COPY TEST1.DAT B:TEST1.ODA
COPY *.OTX B:
KEYPUNCH TEST1.QPL
REM Make back up copy of new data file.
IF EXIST TEST1.DAT COPY YEST1.DAT B:TEST1.NDA
```

In addition to setting KEYPUNCH options, this batch file also demonstrates how you may also ask the interviewer to check the computer's time and date settings and to make backup copies of the data file. This batch program makes two backup copies of the data file. The first backup file will contain a copy of the data file before any changes are made. It has been given an extension of "ODA" to indicate that it is a copy of the old data file. Copies of any open-ended question files are also made before starting the KEYPUNCH program. The second backup file is made after the the KEYPUNCH program has been exited. It is simply a copy of the newest data file, and it will be given the extension "NDA."

Correct Time and Date

The KEYPUNCH program automatically checks to see if the correct date has been set on the computer's internal clock. If the date is less than 1988, the KEYPUNCH program will display an error message and then return control of the computer to the DOS command line without starting the questionnaire. You can also use a batch file, as in the above example, to ask the interviewer to check the computer's time and date before starting the KEYPUNCH program.

Free Disk Space

The KEYPUNCH program also checks to see that there is enough room to store a new record before it will allow a new record to be edited. If there is not enough room, the KEYPUNCH program will display an error message on the screen. If this occurs, the interviewer should quit the KEYPUNCH program, and then begin again using a new formatted disk that contains the KEYPUNCH and your questionnaire programs.

You also should monitor the disks closely if your questionnaire program uses many OPENEND questions. Since the interviewer may enter a large amount of narrative in response to each of these questions, the KEYPUNCH program cannot accurately predict when there will be too little free disk space available to store a new interview. You should check the interviewers' disks regularly to insure that they have enough room for additional interviews (especially if they are using 5¼-inch double-density disks, which hold less information than high-density or 3½-inch disks).⁶

⁶You can use the "Show status," function (F9) on the Options Menu to see how much disk space is available.

Using askSam Text-Based Data Files

askSam describes itself as a “free form information management environment.”⁷ Its ability to easily handle both structured and unstructured information makes it a convenient tool for analyzing the large amount of text information that can be collected in an interview using the OPENEND and STRING questions. You can use askSam interactively to

- search your text data base for interview records that contain specific words or phrases,
- count how often these words were used,
- sort the records,
- extract information from the interviews you find, and
- show the information on the screen, print it, or put it in a file that you can load into your word processor.

In addition to these basic capabilities, you also can use its advanced functions to annotate the information you find, add new information, create hypertext links between related pieces of information in different records, and even create reports that make it easier for new users to query the data file.

Creating an askSam File

Converting your interview data base into an askSam text data base is basically a two-part process: First, you must create a text-based data file using the KEYPUNCH program, and second, you must import this file into askSam for Windows (which will automatically store it in its own format).⁸ The following table lists each step in this process.⁹

⁷ askSam is produced by Seaside Software, Inc., P.O. Box 1428, Perry, Florida 32347, Telephone: 800-800-1997 or 904-584-6590, EMail: info@asksam.com.

⁸ Version 4.0 of the QPL CONVERT program also can create askSam import files. It has additional flexibility that lets you choose which questions to put in the asksam data file. It also does a better job of formatting open-ended responses. The KEYPUNCH conversion to askSam has been kept for backward compatibility with existing projects.

⁹ These steps refer to askSam for Windows, Version 3.0.

Steps to Create an askSam File

1. Start the KEYPUNCH program with your questionnaire program.
2. Print --> Format Options submenu.
3. Check the "Use askSam format" option.
4. Check the "Show open-ended responses" option to include the full text from OPENEND questions. You may also select other format options.¹⁰
5. Close the Format Options menu by pressing the Escape or left arrow keys.
6. Make sure that the option to print to a file, instead of the printer, has been checked.¹¹
7. You are now ready to create a text-based data file. Select one of the first four functions on the Print menu to tell the KEYPUNCH program which interview records you want to put in the data file.¹²
8. Quit the KEYPUNCH program and start askSam for Windows.
9. File --> New...
10. Enter the name of the askSam version of your text-based data file that you want to create. AskSam will always give this file an ASK file name extension.
11. Options --> Word Wrap (Make sure that the Word Wrap option is not checked.)
12. File --> Import...
13. At the Import dialog box, check "Auto Field Recognition."
14. Press the Set Option button.
15. At the Import Options dialog box, set Document Delimiter to "String."
16. Type the delimiter string, "@@".
17. Press OK to close Import Options dialog box.
18. Press OK to begin importing.
19. At the Automatic Field Recognition dialog box, press "Generate List."
20. At the Select Fields dialog box, press "Add All."
21. Press OK to close Select Fields dialog box.

When you use the askSam format option, the KEYPUNCH program will create a large ASCII text file that will look much like the formatted interview records that it normally generates. The format will be changed slightly, however, to accommodate askSam. The variable names you used for each question will be converted into askSam field names by appending a left bracket ([]) to the right side of

¹⁰The most useful options are the "Wrap long string responses" option (if you used many string fields), the "Show each checked response" option (if you used check-all-that-apply questions), and the "Include skipped questions" (if you will be using askSam to write detailed reports). You should not use the "Show first line of question" option because this makes it difficult to use askSam's field searching abilities.

¹¹This option should be checked automatically after you select the askSam format option.

¹²You do not have to select all the records at once. Interview records will always be appended to the end of the KTX file. It is never over-written with new records.

each name. askSam will treat all the information to the right of the bracket as the contents of the field. The KEYPUNCH program will put a closing right bracket (]) at the end of the answer text if it spans more than one line. You can use various askSam commands to search, sort, or extract information from the data file according to the contents of any of these fields.

The KEYPUNCH program also inserts special codes to tell askSam when one interview record ends and another one begins. The characters “@@” are always used to separate interview records. Before you begin importing this file into askSam, you must tell it that these characters have been used to separate interviews, or what askSam calls “documents.”

Running the CONVERT Program

The CONVERT program performs a variety of functions to prepare your questionnaire data for analysis. It can translate your questionnaire program into an SPSS, SAS, Awk, or BASIC program that can read the data files that are created by the COLLECT and KEYPUNCH programs. It defines the card and column location of each variable in the record, assigns variable labels (*i.e.*, the text of each question) and value labels (*i.e.*, the list of answers for multiple choice questions). It also can list the appropriate questions for frequency and descriptive analyses.

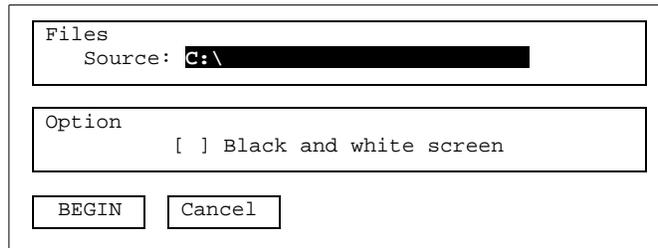
The CONVERT program can also translate the data file that is created by the COLLECT and KEYPUNCH programs into a variety of other data formats. It can translate the data into an askSam import file, Lotus 1-2-3 or Symphony spreadsheet, a dBase database file, or a comma or tab delimited data file. It allows you to pick particular questions and records to keep or drop when translating the data file.

It can also translate the data file of one version of your questionnaire to the record layout of another version. For example, if you decided to add questions to a telephone survey after you had begun using it, the record layouts of the two versions of the survey would not match and could not be analyzed with the same SPSS or SAS programs. The CONVERT program matches the versions of the questionnaire, question by question, and then translates the old data file into the layout of the new file. New questions that did not exist in the old file are set to a missing value code, and questions that were not used in the new file are dropped.

Finally, it can produce a formatted questionnaire document from your program that makes it easy for your clients and interviewers to review your work. It will automatically number the questions, draw boxes for the answers, write skip instructions, print the card and column locations for each response, and divide the document into pages. It also can produce a cross-reference listing of all the questions in your questionnaire by the question's variable name and number. This is useful when debugging your questionnaire program. This listing shows (1) which questions lead to a particular question and (2) all the questions that a particular question may skip to.

There are two ways to tell the CONVERT program which questionnaire you want to use. One way is to use the "Startup Menu," which asks you for the name of your questionnaire and whether you need to set the screen colors to black and white.

Startup Menu



```
Files
Source: C:\
Option
[ ] Black and white screen
BEGIN Cancel
```

The Startup Menu is displayed when you start the CONVERT program without any options, that is, when you start it by typing “CONVERT” and then pressing the Enter key.

Example

```
C:\> CONVERT
```

The Startup Menu lets you edit the file name and set the screen color option before you start the program. A highlight bar indicates which item in the menu you are pointing at. You can move the highlight bar up and down using the arrow keys on the right side of your keyboard or by pressing the Enter key.

The highlight bar always points to the source file name when you first start the program. You must type the name of your compiled questionnaire file in this space. Typing the QPL extension, however, is optional. The CONVERT program will always load the file with this extension, even if you enter a file name with another extension. Use the Insert, Delete, and backspace keys to edit the file name.

Press the space bar to toggle the screen color option on or off. A check mark indicates that an option is on. What the options do is discussed below.

You must press the Enter key to jump from the option to the “Begin” box. Press Enter at this point to start the program, or use the arrow keys to highlight “Cancel” to quit the program. Pressing the up arrow key will take you back to the file names and options. You may also press the Escape key at any time to quit the Startup Menu and return to DOS.

You may instead start the CONVERT program with your questionnaire if you specify its file name at the DOS command line.

DOS Command Line Options

```
CONVERT [filename[.QPL]] [/B] [/H] [/Ifilename.INI] [/L#]

/B      Set screen colors to black and white
/H      Display command line options
/Ifile  Use printer initialization file (default is QPLPRINT.INI)
/L#     Set parallel printer port number (default LPT1:)
```

Note: The brackets indicate optional information. The lowercase words indicate the type of information required. The # character indicates where a number is required.

Example

```
C:\> CONVERT A:SAMPLE
```

In this example, the CONVERT program will use the questionnaire program, SAMPLE.QPL, which is located in the A: disk drive. Note that typing the QPL extension is optional.

Options

As shown above, there are only three options that you can select when starting the CONVERT program. The screen color option is available from both the Startup Menu and the DOS command line. The options to show the DOS command line options or to use a particular printer initialization file are only available from the DOS command line. Other CONVERT options are set using menus after the program has been started. These options are discussed below.

Option /B, Set Screen Colors to Black and White

The CONVERT program displays the program title and error messages on your display screen in color if your computer has a color graphics display card. It automatically switches to black and white if your computer has a monochrome display card, such as a Hercules graphics card. Some computers, however, have a color graphics card connected to a monochrome monitor (many laptop computers are configured this way). In this instance, the CONVERT program would use colors, but they would be unreadable on the monochrome screen. You can force the CONVERT program to use black and white by adding the /B option to the end of the command line when starting the program, or checking “Black and white screen,” on the Startup Menu.

All of the QPL programs will use a black and white screen if you start them with the /B option.

Option /H, Display
Command Line Options

You can tell the CONVERT program to list all of the command line options on the screen if you start it with the /H option. Once the list is displayed, you can press any key to quit the CONVERT program and return to DOS. All the QPL programs will display their command line options if you start them with the /H option.

Option /I, Printer
Initialization File

Printer initialization files are used to program your printer so that it will use a smaller type size and line width. This allows you to print more of your formatted questionnaire document, statistical program, or other types of reports on each page. By default, the CONVERT program will load the initialization file called QPLPRINT.INI if it is located on the same disk and in the same directory as your questionnaire program. You may instead use the /I option to tell the CONVERT program to load another file, that may be located on another disk or in another directory.¹

Option /L#, Set Printer Port

Normally, the CONVERT will print reports on the printer that is connected to your first printer port (*i.e.*, LPT1:). You may use this option to use another port.

Example

```
C:\> CONVERT A:SAMPLE /L2
```

In this example, the /L option was used to tell the CONVERT program to use the printer connected on LPT2: as the default printer.

You may also use the File Menu options to change the printer port after you have started the CONVERT program.

CONVERT Menu

After you have started the CONVERT program, a menu of functions will be displayed across the top of your computer screen. This is called the Main menu. All of the CONVERT functions have been organized under these seven categories in this menu. When you select one of the categories, a second menu will be displayed that shows all of the functions that may be performed in that category.

Main Menu

```
Variables Records Options Data Program File Quit
```

- The Variables menu has functions that allow you to select which questions you want to keep or drop when translating the QPL data file to another format, or

¹For more information on using printer initialization files, see the discussion of printer setup codes in the File menu section at the end of this chapter.

what variables you want to define when generating a BASIC, SAS, or SPSS program.

- The Records menu has functions that allow you to select which records you want to keep or drop when translating the QPL data file to other formats.
- The Options menu can be used to modify how the data translation and the program generation functions work.
- The Data menu lists the types of data translations that may be performed. The CONVERT program will perform the translation when you select it from the menu.
- The Program menu lists types of analysis programs that can be generated. The CONVERT program will write the analysis program when you select it from the menu. The formatted questionnaire document and cross-reference listing of skip instructions are included in the Program menu.
- The File menu provides several different functions that allow you to manipulate the files on your disk. You may look at text files, such as the SAS or SPSS programs, that are generated by the CONVERT program. You also may print, erase, rename, or check the file status of files.
- The Quit menu allows you to exit the CONVERT program and return to the DOS command line.

You must set any options you want to use, such as keeping only certain variables and records, before you select a function to translate the data file or generate an analysis program. These functions always use the current option settings.

Using the Menu

The CONVERT menus are easy to use and are similar to menus that you may have used in other programs, such as Lotus. They allow you to select variables, records, and other options in any order you choose, then translate data or generate an analysis program, then change the options again.

When you first start the CONVERT program, the first item in the main menu, Variables, will be highlighted with a bright background. This highlight bar indicates what function will be performed if you press the Enter key. To choose another item, you must first move the highlight bar. You may move the highlight bar using the arrow keys that are on the right side of your keyboard (Note: The arrow keys will not function properly if the NumLock key is on. That is, if the

keys are set as numbers, instead of arrows.) You can move the highlight bar to the left or right using the left and right arrow keys.

Once you have selected an item from the main menu, a second menu will be displayed directly underneath the item you selected. This is called a “drop-down” menu. It shows all the functions that may be performed for that item. Use the highlight bar to select which function you want to perform, and then press the Enter key to execute that function. If you do not want to perform any of the functions listed in the second menu, press the Escape key to back up to the main menu. The Escape key can always be used to back up or cancel any function that you have started.

Menu Key Summary

Left	Move highlight bar to the left
Right	Move highlight bar to the right
Up	Move highlight bar up
Down	Move highlight bar down
Enter	Execute highlighted menu item
Esc	Back up to main menu or cancel function
Spacebar	Toggle an option check mark on and off

Once you become familiar with the CONVERT menus, you may want to use the special “hot” keys to speed up your menu selections. When the highlight bar is on the main menu, you can type the first letter of one of the item names to automatically select and display its menu. When the highlight bar is in one of the drop-down menus, you can type the first letter of an item to make the highlight bar jump directly to that item. The function is not executed, however, because the names of more than one item may begin with the same letter.

You may also use the function keys to start certain functions. If a function can be started with a function key, the name of the function key will be shown on the drop-down menu. The function keys allow you to automatically jump to another drop-down menu and execute a function.

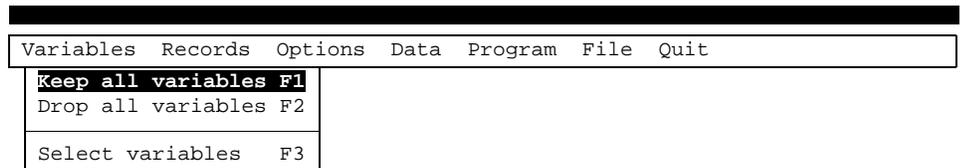
CONVERT Function Keys

F1	Keep all variables
F2	Drop all variables
F3	Select variables to keep or drop
F4	Rename a disk file
F5	Select records to keep or drop by numeric search
F6	Select records to keep or drop by string search
F7	Quit CONVERT program
F8	Erase a disk file
F9	Show status of a disk file
F10	View a text file from a disk

Variables Menu

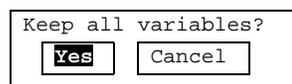
The Variables menu allows you to select which variables (*i.e.*, questions) will be kept or dropped when the CONVERT program makes Lotus, dBase, or free format files, or what questions will be defined when it generates an Awk BASIC, SAS, or SPSS program. The first two items in the menu, “Keep all variables” and “Drop all variables,” allow you to set the status of all the variables in your questionnaire at once. When you first start the CONVERT program, all of the variables will be set to keep. It is sometimes useful, however, to reset them all to drop if you are going to keep only a few of them. In this case, you can select the few you want to keep using the last item in the Variables menu, “Select variables.”

Variables Menu



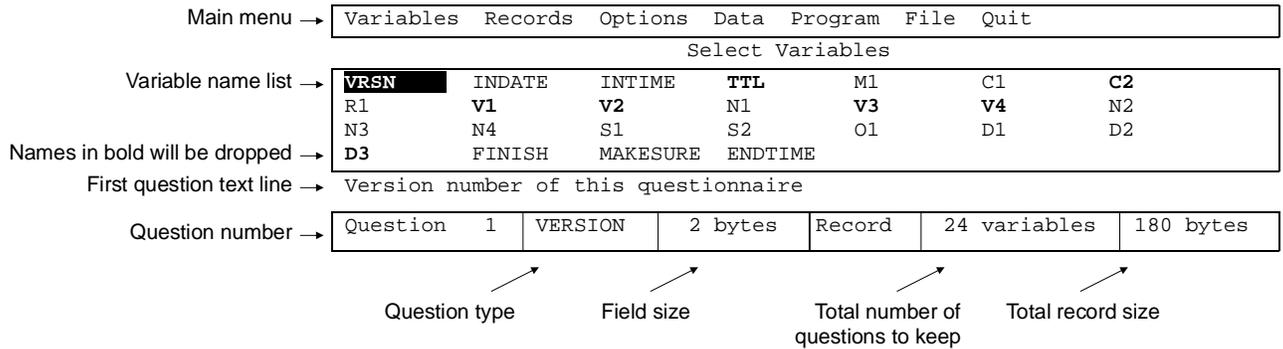
If you select the keep or drop function, a dialogue box will be displayed that asks you to confirm this action. You can use the arrow keys to highlight your answer and then press Enter to execute the function. You may also press the Escape key to cancel this function.

Variables Menu



If you choose the select variables function, a large dialogue box will be displayed that lists all the variables in your questionnaire, and summary information on each variable, and information on how many variables are set to be kept and the total record size of those variables.

Variables Menu



When you first display this dialogue box, the highlight bar will be on the first variable in your questionnaire. Also, the text of the question for this variable is shown at the bottom of the screen, as well as its question number, type, and size. You can move the highlight bar to other variable names using the arrow keys. You also may type the first letter of a variable name to jump to the next question that starts with that letter. Finally, you can display the full text of the highlighted question by pressing the F2 key, or a summary of all the key functions by pressing the F1 key.

All variables that are set to be dropped (and VOID questions that do not store any data) will be displayed in bold type if you are using a monochrome monitor or in your highlight color if you are using a color monitor. You may toggle variables from keep to drop, or drop to keep, by first moving the highlight bar to that question's variable name and then pressing the Enter key or the space bar. The variable name will switch from normal to bold, or vice-versa, to indicate whether it will be kept or dropped. The summary information shown at the bottom of the screen will also be updated to show the total number of variables that will be kept and their total record size.

Press the Escape key to return to the Main menu when you have finished selecting variables.

Select Variables Dialogue Keys

arrow keys	Move highlight bar up, down, left, and right
letter	Move highlight bar to the next name that starts with this letter
Home	Move highlight bar to the first name
End	Move highlight bar to the last name
Enter or space bar	Toggle variable between keep and drop
Esc	Finish selection and return to main menu
F1	Display key function summary
F2	Display full text of highlighted question

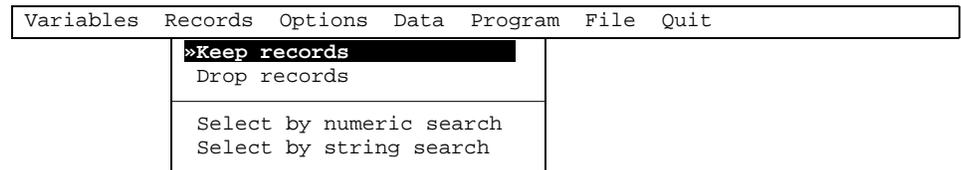
Records Menu

The Records menu can be used to specify which interview records will be kept or dropped when the CONVERT program translates your QPL data file to

- askSam,
- Lotus,
- dBase, or
- a comma or tab delimited file.

It also affects which records are copied when you translate your QPL data file to the record layout of a newer version of your questionnaire, when you copy the file to another disk or directory, and when you convert the OPENEND question OTX files into a single ASCII text file. Thus, this function gives you the ability to easily split your data file into smaller parts, or into files that have only certain types of records, at the same time you are converting it to another format.

Records Menu



The Records menu has four items. The first two, “Keep records” and “Drop records,” tell the CONVERT program what to do if your selection criteria are true for a record. A check mark next to the item name indicates which item has been selected. It is always set to keep records when you first display the menu. You may change it to drop records by moving the highlight bar to “Drop records” and then pressing the Enter key. Using the drop option sometimes simplifies your selection criteria. It allows you to avoid having to use “not equal” specifications, which can become confusing and therefore prone to errors.

There are two ways to select records. You may write an expression, using numeric variables, that will evaluate to a true or false value to indicate whether a particular record should be kept or dropped. You also may identify records by searching for a word or phrase. If a match is found, then the record will be either kept or dropped. A check mark will be displayed on the menu if you have entered an expression or search string.

You may use both specifications at the same time. In this case, they both must evaluate to a true value in order for a particular record to be kept or dropped. If you do not use any record specifications, the CONVERT program will drop or keep every record, depending upon which you have selected.

Select by Numeric Search

When you select this function, a dialogue box will be displayed in the middle of the screen that asks you to enter a search expression. You may enter any valid QPL expression that, when evaluated for each record, results in a true or false value. You can refer to any numeric questions by their variable name. You can also refer to them by their question number if you precede the number with a “#” sign.

Example

```
INDATE < 19890701 AND #3 < 112
```

In this example, the record selection criteria have been based on the response to two questions: INDATE and question number three. Here, a record will be kept or dropped if INDATE is earlier than July 1, 1989, and if the answer to question number three is less than 112.

You also may use the keyword RECORD in an expression to identify interviews by their record number (*i.e.*, their physical location in the data file).

Example

```
RECORD <= 10
```

In this example, the RECORD keyword will be used to drop or keep the first 10 records in the data file.²

If you have forgotten the name of a question, you can display a list of all the questions by pressing the F2 key. After opening the variable list dialogue box,

²See chapter 4 for more information on how QPL expressions are evaluated.

you can automatically insert the name of a question into your expression by first highlighting it and then pressing the Enter key. The dialogue box will close and the name will be inserted at the current cursor position. You can close the dialogue box without copying the name of a question by pressing the Escape key.

Press the Enter key to return to the Records menu after you have typed in your expression. You can quit the numeric search dialogue box by pressing the Escape key or by entering a blank expression. A check mark on the Records menu indicates whether an expression will be used to select records.

If you want to delete or change an expression that you have already entered, simply select the numeric function again and edit the expression using the backspace, Delete, and Ctrl-End (*i.e.*, delete from the cursor to the end of the line) keys.

Select by String Search

When you select this function, a dialogue box will be displayed in the middle of the screen that asks which question you want to search and the phrase you want to search for. You may perform a string search on any type of question, except VOIDS, but you must write the search key using the format with which the response is stored in data file. Typically, this function is used to search for responses to STRING questions.

You may enter either the variable name of the string question or its number. (Note: The “#” is not required.) If you have forgotten the name of the question, you may press the F2 key to display a list of question names. You may copy one of the names to the string search dialogue box by pressing the Enter key. After you have entered the variable name, press the Enter key to begin entering the search phrase.

The search phrase may be any combination of letters and numbers. Upper and lowercase letters are treated equivalently during the search. For example, the word “automobile,” will match the word “AUTOMOBILE.”

Example

```
Variable: S1  
Phrase: computer
```

In this example, a record will be kept or dropped if the CONVERT program finds the phrase “computer system” or “MAINFRAME COMPUTER” in an answer to question S1.

Options Menu

Variables	Records	Options	Data	Program	File	Quit
		Redefine long strings to short				
		Define date components				
		Write variable names at top				
		Set missing numbers to -1				
		Numbered variable labels				
		Frequency statistics				
		Descriptive statistics				
		List long strings				
		80 column page				
		»132 column page				

Press the Enter key to return to the Records menu after you have typed in your search phrase. You can quit the string search dialogue box by pressing the Escape key or by not entering a variable name. A check mark on the Records menu indicates whether a search phrase will be used to select records.

To delete or change a search phrase that you have already entered, select the string search function again and edit the variable name and phrase using the backspace, Delete, and Ctrl-End (*i.e.*, delete from the cursor to the end of the line) keys.

Options Menu

The Options menu contains all of the other options that can be used to modify how the CONVERT program will translate data files and generate analysis programs. Each of these options is specific to one or more of the data translation and program generation functions. Which functions each option is used with will be displayed at the bottom of the screen as you move the highlight bar up and down the list.

The purpose of each of these options will be discussed in the following two sections on the Data and Program menus.

An option is on if a check mark is displayed. You can toggle the option on or off by moving the highlight bar to that option and then pressing the Enter key or space bar. Press the Escape key to return to the main menu after you have finished selecting your options.

Data Menu

The Data menu contains all of the functions to translate the data file that is created by the COLLECT and KEYPUNCH programs into other formats, such as Lotus and dBase, and into the record layout of a newer version of your questionnaire. You must select any options you want to use, such as setting which records you want to keep and drop, before you select one of these functions. They will always use the current option settings when translating QPL data files.

Data Menu

Variables	Records	Options	Data	Program	File	Quit
			ATX	askSam	import	
			DBF	dBase III+		
			DBF	dBase IV		
			DAT	Copy data file		
			DAT	Translate to new QPL program		
			CTX	Convert OTX files to ASCII		
			PRN	Comma delimited		
			PRN	Tab delimited		
			WKS	Lotus Version 1A		
			WK1	Lotus Version 2.01		
			WR1	Lotus Symphony		

When you select one of these functions, a dialogue box will be displayed in the middle of the screen that asks what name you would like to give to the data file that it is about to create. It always suggests a default name that you may use. The default name is the same as your QPL program (including the disk drive and directory), but it will have an extension that is appropriate for the type of file that you are creating. These extensions are shown on the left of each line of the menu. You may change the name, including the disk drive and directory information, using the backspace, Delete, and Ctrl-End (*i.e.*, delete from the cursor to the end of the line) keys. Press the Enter key after you have chosen a file name. Select “Begin” to start translating the data file or “Cancel” to abort this function without translating any data. You may also press the Escape key at this point to quit the translation function and return to the Data menu.

Data Dialogue Box

Variables Records Options Data Program File Quit

Enter dBase file name:

Once you begin translating the data, the number of the record that is currently being processed and the total number of records that have been put into the new data file will be displayed in the middle of the screen.

askSam Import Files

AskSam describes itself as a “free form information management environment.”³ Its ability to easily handle both structured and unstructured information makes it a convenient tool for analyzing the large amount of text information that can be collected in an interview using the OPENEND and STRING questions. You can use askSam interactively to

- search your text data base for interview records that contain specific words or phrases,
- count how often these words were used,
- sort the records,
- extract information from the interviews you find, and
- show the information on the screen, print it, or put it in a file that you can load into your word processor.

In addition to these basic capabilities, you also can use its advanced functions to annotate the information you find, add new information, create hypertext links between related pieces of information in different records, and even create reports that make it easier for new users to query the data file.

This function will translate your QPL data, including the responses to open-ended questions,⁴ to an ASCII text file that is specially formatted for easy loading and use by askSam. CONVERT will automatically copy all the data from the fields you have specified in the Variables menu and all the records you specified in the Records menu. The same variable names you used in your questionnaire will be

³askSam is produced by Seaside Software, Inc., P.O. Box 1428, Perry, Florida 32347, Telephone: 800-800-1997 or 904-584-6590, EMail: info@asksam.com, World Wide Web: <http://www.asksam.com>.

⁴The question text that is embedded within each response (OTX) file is also stripped out, leaving just the response, when the askSam import file is created.

used as field names in the askSam import file. The QPL data file record number will always be written to the first field and will be given the name "RECORD[."

The variable names you used for each question will be converted into askSam field names by appending a left bracket ([) to the right side of each name. AskSam will treat all the information to the right of the bracket as the contents of the field. The CONVERT program will put a closing right bracket (]) at the end of the answer text. You can use various askSam commands to search, sort, or extract information from the data file according to the contents of any of these fields.

The CONVERT program also inserts special codes to tell askSam when one interview record ends and another one begins in the import file. The characters "@@" are always used to separate interview records. Before you begin importing this file into askSam, you must tell it that these characters have been used to separate interviews, or what askSam calls "documents."

Only the Variables and Records menu functions, as mentioned above, affect how an askSam file is created. The Options menu does not affect how an askSam import file is created.

You must use askSam's File -> Import... function to load this file into askSam. The following table lists the steps⁵ necessary to import the file that CONVERT creates into askSam for Windows.

Loading an Import File Into askSam

1. Start askSam for Windows.
2. File -> New...
3. Enter the name of the askSam version of your text-based data file that you want to create. AskSam will always give this file an ASK file name extension.
4. Options -> Word Wrap (Make sure that the Word Wrap option is checked.)
5. File -> Import...
6. At the Import dialog box, check "Auto Field Recognition."
7. Press the Set Option button.
8. At the Import Options dialog box, set Document Delimiter to "String."
9. Type the delimiter string, "@@".
10. Press OK to close Import Options dialog box.
11. Press OK to begin importing.
12. At the Automatic Field Recognition dialog box, press "Generate List."
13. At the Select Fields dialog box, press "Add All."
14. Press OK to close Select Fields dialog box.

⁵These instructions refer to askSam for Windows, Version 3.0.

dBase III+ and IV Files

These functions will translate a QPL data file into a dBase III+ or IV “DBF” file. CONVERT will automatically define all the fields you have specified in the Variables menu and copy all the records you specified in the Records menu. You may put up to 127 questions in a dBase III+ file and up to 254 questions in a dBase IV file.⁶ The same variable names you used in your questionnaire program will be used in the dBase file. The QPL data file record number will always be written to the first field and given the name “RECRD.” This file can be directly loaded into dBase with its USE command.

Redefine Long Strings to Short

There are two options, in addition to setting the variables and records to be copied, that can be used to modify how the dBase file is created. The first option relates to how string questions are converted to dBase character fields. Normally, if the string field is less than 254 characters long (the dBase maximum field size), the whole answer will be copied into a dBase character field. If it is longer than this, then just the first 254 characters will be copied.

The “Redefine long strings to short” option on the Options menu can be used to further truncate string questions. If this option is checked (*i.e.*, turned on), the CONVERT program will truncate any string questions that are longer than 8 characters. This option may be useful when you need to shrink the data record.

Set Missing Numbers to -1

The QPL COLLECT and KEYPUNCH programs set answers to questions that were skipped, and answers to numeric and date questions that were not known, to a missing value code (*i.e.*, blank spaces) when they save each record. dBase, however, does not handle missing data this way. A number must always be entered into each numeric field. The CONVERT program will normally change all missing number values to zero when it copies the data to a dBase record.

This may not always be convenient, however, because zero may be a valid answer to some questions. In this situation, you would not be able to tell the difference between a skipped, or not known, response and a zero response. You may instead use the “Set missing numbers to -1,” option from the Options menu to tell the CONVERT program to change missing numbers to -1. This allows you to distinguish between missing and not missing information since negative numbers are rarely entered as a response to a question.

Copy Data File

The “Copy data file” function allows you copy all or part of your file of interview records. You can selectively drop or keep records by specifying a search expression or string in the Records menu. Thus, you could use this function to delete records from your data file by copying the ones you want to another file

⁶Each answer in a check-all-that-apply question will be counted as a separate dBase variable.

and then renaming the files. All the records will be copied if you do not specify any selection criteria.

This function does not, however, allow you to delete variables from each record. To do this, you must delete questions from a new copy of your original questionnaire program, recompile it, then translate the data file from the original record layout to the record layout of the new program. The next section describes how to do this.

When you select this function from the menu, a dialogue box will be displayed in the middle of the screen that asks for a name for the new data file. You must change the default name, which is the name of the current data file, because you may not copy a file to itself. You may, however, use the same file name if you change the disk drive or directory specifications.

Appending Records

If you specify a new file name, the CONVERT program will open a new file with that name, on that disk and in that directory, and proceed to copy all of the records that meet your specifications. If you specify the name of an existing data file, the CONVERT program will open that file and append all of the records that meet your specifications to the end of the file.

You may work with the new data file with the CONVERT program if you rename the two data files so that the new file has the same name as your QPL program, and the original data file has a new name. The CONVERT program will then use the new data file in any further processing you may wish to do, such as creating a Lotus spreadsheet. Files may be renamed using the “Rename file” option located in the Files menu.

Translate Data File

The “Translate to new QPL program” option allows you to convert a data file that is in the record layout of one version of your questionnaire into the record layout of a newer version of your questionnaire. This function is useful when you have already begun to enter data from interviews using the COLLECT or KEYPUNCH programs, but decide to change your questionnaire program to handle an unforeseen problem. For example, if you have to add a question, the record layout of the new questionnaire program will not match the record layout of the old program, and thus the data collected with both programs cannot be analyzed with the same analysis program.

The translate function solves this problem by matching the questions, according to their variable names, and copying the data, variable by variable and record by record, from the old file to the record layout in the new file. The translate func-

tion will translate a data file if at least one of the questions in both files have matching variable names.

The translate function can accommodate most changes in a record layout. There are certain precautions you should take, however, before using it:

- Make sure you have backup copies of both data sets. If you make a mistake, you can still recover if you still have a copy of the original data.
- Do not change the names of questions. The translate function matches questions by their names. If it does not find a match, it will drop the answers to that question during the translation.
- Do not change a question's TYPE. Multiple choice questions, for example, cannot be translated into numeric questions.
- Do not change the order of the answers to multiple choice, check-all-that-apply, or random number questions. These answers are coded according to their position in the list. If you change the order, the codes from the old questionnaire will not have the same meaning as the codes for the new questionnaire. If you need to add responses, you should put them at the bottom of the list.

Translation Rules

The translation function will perform a variety of operations on the data in order to make them fit a new record layout. The following list summarizes some of the more common translation problems and how the translate function handles them.

- Old question is not used in the new questionnaire: Old variable is not copied.
- New question did not exist in the old questionnaire: New variable field is filled with the missing value code (*i.e.*, blank spaces).
- Question type was changed in new questionnaire: Old variable is set to missing value code.
- String question was lengthened or shortened: Spaces are added or deleted from the right end of the string.
- Number of decimal places were added or deleted from a number: Zeros are added to right side, unless number is missing. Then the field is filled with the missing value code. If deleted, digits are dropped from the right side.

- Number of digits were added or deleted from a number: Spaces are added to the left side if digits were added; characters dropped from the left if they were deleted.

As you can tell from these two lists of rules, there is a lot to consider when translating files. The most important rule, however, is to make backup copies of your data files before using this function. The second most important rule is to always inspect the new data file for translation errors with the KEYPUNCH program before using it for analysis.

When you select this function from the menu, a dialogue box will be displayed in the middle of the screen that asks for the name of your new QPL program. You must change the default name, which is the name of the current QPL program, because you may not translate a file to itself.

If your new program does not have a data file, the CONVERT program will open a new data file, on the same disk and directory, and proceed to translate all of the records that meet the specifications you set in the Records menu. If your QPL program already has a data file, the CONVERT program will open that file and append all of the records that meet your specifications to the end of that file.

Comma and TabDelimited Files

The CONVERT program will copy all of the variables you specified in the Variables menu, for all of the records that meet the specifications you set in the Record menu, and translate them to a comma or tab delimited file (using the “PRN” file name extension). These data files can be easily read by many different types of analytical software packages and general purpose programming languages. You may want to create this type of file if you are using analytical software that is not supported by the CONVERT program.

These files, also called free format files, are not defined by the column position of variables, but instead by the order that variables appear in the record. Individual variables are separated from each other by commas or tab characters (ASCII code 9). The string variables in comma delimited files are also surrounded with double quotes. Only one record is on each line, and each line is terminated by carriage return and line feed codes.

Example

Question	Comma Delimited File Format
1. Date: February 10, 1987	1,870210,28800,"ABC",30.5,1
2. Time: 8:00 a.m.	- ===== -
3. String: ABC	^a 1 2 3 4 5
4. Number: 30.5	
5. Multiple Choice: Yes (First answer in list)	Tab Delimited ^b File Format

	1^870210^28800^ABC^30.5^1
	- ===== -
	^a 1 2 3 4 5

^aThe record number of each case is always written to the first position of the delimited file.

^bThe caret (^) indicates where a tab character is located.

This file structure, however, can also cause problems. Since all of the variables are written to one line, the lines tend to become very long, and may exceed the capacity of your software package to read them. Lotus, for example, cannot read delimited files that have line lengths greater than 240 characters, including commas (using its “/File Import Numbers,” command). If you run into this problem, you must split your data file into several files by copying only a few variables at a time.

Options

There are several options available that you can use to modify how these functions work, in addition to specifying the variables and records you want to copy. These options must be set on the Options menu before you use this function.

Redefine Long Strings to Short

The CONVERT program trims blank spaces from both ends of a string field and then adds double quotation marks to each end of the string before it copies the answer to a delimited file. The entire answer will be copied to the new data file, no matter how long it is. You may use the “Redefine long strings to short,” option to truncate strings that are longer than 8 characters.

Define Date Components

The CONVERT program normally copies dates to delimited files as one number, in YYMMDD or YYYYMMDD format, according to the type of date you have used. You may instead tell the CONVERT program to write dates as separate numbers if you check the “Define date components” option. The numbers will still be in year-month-day order, but the numbers will be separated by commas. In this case, you must define one date variable as three variables in your analysis program that reads the delimited file.

Example

Date	Components Not Defined	Components Defined
----- January 3, 1989	----- 890103	----- 89,1,3

Write Variable Names At Top

Some application programs will also let you define the names of the variables in a comma delimited file. You can use the “Write variable names at top,” option to tell the CONVERT program to write all of the names of the variables you have selected to copy in the first line of the file.

Example

```
Comma Delimited
File Format
-----
"RECORD", "INDATE", "INTIME", "S1", "N1", "M1"
1,870210,28800,"ABC",30.5,1

Tab Delimited
File Format
-----
RECORD^INDATE^INTIME^S1^N1^M1
1^870210^28800^ABC^30.5^1
```

Note: The caret (^) indicates where tab characters are located.

This example shows how the earlier example would be written if variable names option was used.

Set Missing Numbers to -1

Answers to questions that were skipped, or entered as “Don’t know” responses by the interviewer, are set to a missing value code (*i.e.*, blank spaces) in the data file created by the COLLECT and KEYPUNCH programs. Normally, missing numbers in the comma delimited file are indicated only by a comma, without any preceding number. Some analytical software packages may not, however, recognize a lone comma as a missing value. These packages typically skip over the comma and use the next number on the line as the value for that variable, and thus misread your data.

You may remedy this situation by using the “Set missing numbers to -1” option. This option tells the CONVERT program to replace any missing numbers it finds with a value of -1. Then you can program your analytical software package to treat -1 as a missing value.

Example

```
Current Answers
-----
1. Multiple choice: Yes (first answer)
2. Number: 35.6
3. Date: January 3, 1989
4. String: ABCD
5. Multiple choice: Skipped
6. Number: Not known
7. Date: Not known
8. String: Skipped

Normal Comma Delimited Record
-----
1,1,35.6,890103,"ABCD",,,," "

Comma Delimited Record With Missing Numbers Set to -1
-----
1,1,35.6,890103,"ABCD",-1,-1,-1," "
```

Note: The first number in each record line is the record number.

This example shows how data from four types of questions will be written to a comma delimited data file, with and without replacing missing numbers with -1. Empty or missing strings are not a problem because the quotation marks always indicate where string data are located on the line.

Tab delimited files are usually read properly without changing the missing values to -1. If they are not read correctly by your analysis application, you should recreate your tab delimited data file using the -1 option as discussed above.

Lotus Files

The Lotus functions allow you to translate your data into Lotus 1-2-3 Version 1A or 2.01, or Symphony spreadsheets. These functions also do much of the work of formatting the spreadsheet. For example, these functions will

- write a documentation box at the top of the spreadsheet,
- set the column width for each variable,
- set the number of decimal places to display,
- use a currency format if you used one in your source QPL program, and
- use the @DATE function and the date format to store and display dates.

You may specify which variables and records to copy to the spreadsheet using the Records and Variables menus. Each record will be written to one row of cells, and each column will hold all the data for one variable. You can copy up to

254 variables to a spreadsheet.⁷ If you have more than this, you will have to drop some using the Variables menu. The number of each record will always be written to the first column.

Options

There are several options you can use to modify how these functions create spreadsheets. You must set these options, using the Options menu, before you select one of these functions to create a Lotus spreadsheet.

Redefine Long Strings to Short

The CONVERT program trims blank spaces from both ends of a string field before it copies the answer to a Lotus spreadsheet. The entire answer, up to a maximum of 240 characters (the maximum that Lotus can store), will be copied to the spreadsheet. You may use the “Redefine long strings to short” option, however, to truncate strings that are longer than 8 characters.

Define Date Components

Normally, the CONVERT program will copy dates to spreadsheets as a single number in YYMMDD format. You may instead tell it to copy dates using the Lotus @DATE function, and display them using a date format, if you check the “Define date components” option on the Options menu.

Example

Date	Date Components Not Defined	Date Components Defined	Displayed in Date Format
January 3, 1989	890103	@DATE(89,1,3)	3-Jan-89

The Lotus @DATE function does not allow the year component to be less than zero or greater than 99 (corresponding to dates from 1900 to 1999). To accommodate this limitation, the CONVERT program will drop the century portion of LDATE (8-digit date) questions when it copies them to a Lotus file. For example, the date January 1, 2010, will be put into a Lotus date function as @DATE(10,1,1), which Lotus will interpret as January 1, 1910. Thus, you should make sure that none of your dates are earlier than 1900, or later than 1999, before using the Lotus @DATE function.

Write Variable Names at Top

If you are copying many variables to a spreadsheet, you should use the “Write variable names at top” option to label each column of data with the name of the variable it contains.⁸ Checking this option also tells the CONVERT program to define range names for each column of data. The range names will be the same

⁷Each answer in a check-all-that apply question is counted as a separate variable.

⁸The number of each check-all-that-apply answer will be appended to the question name at the top of the column.

as the variable names. You can see that the ranges have been set using the Lotus “/Range Names Create” command.

Set Missing Numbers to -1

The CONVERT program automatically skips over a cell when the number for that cell is missing, according to your program’s skip instructions, or when it is set to a not known value by the interviewer. This normally does not cause any problems when working with the spreadsheet because many Lotus functions, such @SUM and @AVG, will correctly treat blank cells as not a member of the list.

Some functions, such as the @IF function, however, treat blank cells and cells containing a value of zero equivalently. You cannot use the @IF function to distinguish between cells with a valid value of zero and cells where no value was entered. You must use the “Set missing numbers to -1” option to remedy this problem. Checking this option tells the CONVERT program to replace any missing values with -1 as it creates a spreadsheet.

List Long Strings

Normally, the Lotus conversion functions copy the number of the response to multiple choice (MULT) and check-all-that-apply questions (CHECK) to the spreadsheet file. Selecting this option tells the CONVERT program to instead copy the label of the answer that was selected. For example, if a respondent answered “Yes” to a MULT question, the word “Yes” would be put in the spreadsheet file instead of the number one (1).

Program Menu

The Program menu contains functions that can translate your questionnaire program into an Awk, BASIC, SAS, or SPSS program that can read the data file created by the COLLECT and KEYPUNCH programs. The CONVERT program generates the source code (*i.e.*, text of the program) for each of these programming languages. This allows you to load these program files into your programming editor to further customize how they work. You must, however, have one of these software

Program Menu

Variables	Records	Options	Data	Program	File	Quit
AWK Thompson AWK						
BAS Microsoft QBasic						
SAS SAS/PC						
SPS SPSS for Windows						
SPC SPSS/PC+						
SPX SPSS-x						
TXT Formatted question list						
XRF Cross reference list						

packages in order to run one these programs. For example, you must have SPSS software installed on your computer in order to run the SPSS program that CONVERT creates.

The functions that create a formatted questionnaire document and a cross-reference listing of your source program also are included in the Program menu.

You must select any options you want to use, such as setting which variables you want to define in a program, before you select one of these functions. They will always use the current option settings when generating an analysis program.

When you select one of these functions, a dialogue box will be displayed in the middle of the screen that asks what name you would like to give to the program file that it is about to create. It always suggests a name that you may use. The default name will be the same as your QPL program (including the disk drive and directory), but it will have the extension that is appropriate for the file that will be created. These extensions are shown on the left of each line of the menu. You may change the name, including the disk drive and directory information, using the backspace, Delete, and Ctrl-End (*i.e.*, delete from the cursor to the end of the line) keys. Press the Enter key after you have chosen a file name. Select “Begin” to start creating a program or “Cancel” to abort this function without creating a program. You may also press the Escape key at this point to quit the translation function and return to the Program menu.

Thompson Awk Programs

This function creates a Thompson Awk program that can read the data file that is created by the COLLECT and KEYPUNCH programs.⁹ You must edit this program to make it do something in particular with the data.

The Awk program function will, by default, create a program that defines all of the questions in your questionnaire. It also will use the same variable names that you used in your questionnaire. You may use the Variables menu functions, however, to pick which questions will be defined in the Awk program. Selecting only the questions you need to use for your work will make the Awk program run faster and use less memory.

⁹Thompson Awk is an MS-DOS version of the standard Unix Awk interpreter. The Awk programs CONVERT generates may be run under other versions of Awk the support the Thompson extensions to the language (the most important extension being multi-dimensional arrays). The Thompson Awk interpreter and the compiler, called TAWK, may be obtained from Thompson Automation, Inc., 5615 SW Jefferson, Portland, OR 97221; Telephone: (503)224-1639.

You may also select options that tell CONVERT to add routines to the program to do basic analysis of the data including frequency and descriptive statistics, and listing responses to short answer questions. This capability allows you to create an executable program (if you compile it with the TAWK compiler to create an EXE file) that you may freely distribute with your questionnaire application.¹⁰ While these routines do not have the statistical sophistication of SPSS or SAS (no cross-tabulations, for example), you may find them useful for preliminary peeks at your data or peeks at locations where you do not have SPSS or SAS installed.

Example

```
#####
*** PROGRAM: A:\TEST1.AWK                      DATE: 1-31-1996 **
*** AUTHOR:                                     JOB:          **
***                                           **
*** TITLE: QPL Demonstration                    **
***                                           **
*** PURPOSE:                                    **
***                                           **
***                                           **
***                                           **
***                                           **
#####
BEGIN      {   print "QPL Demonstration";
              print "Version 4.0";
              print "Job processed on: " ctime(time())
              print "QPL Demonstration" > stderr;
              print "";
              Card = 0;
              LastCard = 3;
              Case = 0;
            }

            #Count data cards
            {   Card++;
              if (Card > LastCard){
                Card = 1;
              }
            }

            #Read data
Card == 1   {   VRSN = Trim(substr($0, 1, 2));
              INDATE = Trim(substr($0, 3, 6));
              INTIME = Trim(substr($0, 9, 5));
              M1 = Trim(substr($0, 14, 1));
              C101 = Trim(substr($0, 15, 1));
              C102 = Trim(substr($0, 16, 1));
              C103 = Trim(substr($0, 17, 1));
              C104 = Trim(substr($0, 18, 1));
              C105 = Trim(substr($0, 19, 1));
              C106 = Trim(substr($0, 20, 1));
              C107 = Trim(substr($0, 21, 1));
              R1 = Trim(substr($0, 22, 1));
              N1 = Trim(substr($0, 23, 4));
```

¹⁰This differs from SAS or SPSS which are only interpreters. Users must have access to the SAS or SPSS software in order to run the programs that CONVERT creates or your own programs.

```
        N2 = Trim(substr($0, 27, 5) "." substr($0, 32, 2));
        N3 = Trim(substr($0, 34, 5) "." substr($0, 39, 2));
        N4 = Trim(substr($0, 41, 5) "." substr($0, 46, 2));
    }

Card == 2    {    S1 = Trim(substr($0, 1, 100));
                S2 = Trim(substr($0, 101, 7));
                O1 = Trim(substr($0, 108, 8));
            }

Card == 3    {    D1 = Trim(substr($0, 1, 6));
                D2 = Trim(substr($0, 7, 5));
                FINISH = Trim(substr($0, 12, 1));
                MAKESURE = Trim(substr($0, 13, 1));
                ENDTIME = Trim(substr($0, 14, 5));
            }

        #Tabulate data
Card == LastCard    {    Case++;
                        printf("\b\b\b\b\b\b%5d", Case) > stderr;
# Insert your program statements here.
                    }

        #Print summary data
END    {    print "Total cases: " Case;
           print "";

           print "" > stderr;
        }

function Trim(String) {
    sub(/^[\ ]*/, "", String);
    sub(/[ ]*$/, "", String);
    return String;
}
```

Note: This program, TEST1.AWK, was generated from the questionnaire program called TEST1.QPL, which has been included on the QPL Master Program Disk.

This example is an Awk program that was generated from a short questionnaire. This program only reads the data file that was created by COLLECT or KEYPUNCH. You must edit the program to make it do something useful. You should add your program statements at the comment, "Insert your program statements here."

Microsoft QBasic Programs

This function creates a QBasic program that can read the data file that is created by the COLLECT and KEYPUNCH programs. You can load this program into the QBasic interpreter, and then modify it to perform specific tasks with the data.^{11,12} The QBasic program will define your data file using the MID\$ function and pro-

¹¹QBasic is the BASIC interpreter sold by Microsoft with Versions 5 or 6 of MS-DOS.

¹²QBasic syntax is generally compatible with Microsoft's Visual Basic. You may cannablize the QBasic program that CONVERT generates for use in a Visual Basic program.

vide functions that convert numeric data from strings (the way the data are stored in the data file) to integers, and single or double precision real numbers.

Example

```
*****  
'** PROGRAM: A:\TEST1.BAS DATE: 1-31-1996 **  
'** AUTHOR: JOB: **  
'** ** **  
'** TITLE: QPL Demonstration **  
'** ** **  
'** PURPOSE: **  
'** ** **  
'** ** **  
'** ** **  
'** ** **  
'** ** **  
'** ** **  
'** ** **  
*****  
'** Define data conversion functions.  
DEF FNDOUBLE#(STRFLD$,FLDLEN%,FLDDEC%) = CDBL(VAL(LEFT$(STRFLD$,  
FLDLEN%-FLDDEC%)+". "+RIGHT$(STRFLD$,FLDDEC%)))  
DEF FNSINGLE!(STRFLD$) = VAL(STRFLD$)  
DEF FNINTEGER%(STRFLD$) = CINT(VAL(STRFLD$))  
'** Open data file and read data.  
OPEN "A:\TEST1.DAT" FOR INPUT AS #1  
LastCard = 3  
Card = 0  
CaseCount = 1  
DO WHILE (NOT(EOF(1)))  
LINE INPUT #1, InCard$  
Card = Card + 1  
IF (Card > LastCard) THEN  
Card = 1  
CaseCount = CaseCount + 1  
END IF  
SELECT CASE Card  
CASE 1  
VRSN$ = LTRIM$(RTRIM$(MID$(InCard$, 1, 2)))  
INDATE! = FNSINGLE!(MID$(InCard$, 3, 6))+19000000  
INTIME! = FNSINGLE!(MID$(InCard$, 9, 5))  
M1% = FNINTEGER%(MID$(InCard$, 14, 1))  
C101% = FNINTEGER%(MID$(InCard$, 15, 1))  
C102% = FNINTEGER%(MID$(InCard$, 16, 1))  
C103% = FNINTEGER%(MID$(InCard$, 17, 1))  
C104% = FNINTEGER%(MID$(InCard$, 18, 1))  
C105% = FNINTEGER%(MID$(InCard$, 19, 1))  
C106% = FNINTEGER%(MID$(InCard$, 20, 1))  
C107% = FNINTEGER%(MID$(InCard$, 21, 1))  
R1% = FNINTEGER%(MID$(InCard$, 22, 1))  
N1# = FNDOUBLE#(MID$(InCard$, 23, 4), 4, 0)  
N2# = FNDOUBLE#(MID$(InCard$, 27, 7), 7, 2)  
N3# = FNDOUBLE#(MID$(InCard$, 34, 7), 7, 2)  
N4# = FNDOUBLE#(MID$(InCard$, 41, 7), 7, 2)  
CASE 2  
S1$ = LTRIM$(RTRIM$(MID$(InCard$, 1, 100)))  
S2$ = LTRIM$(RTRIM$(MID$(InCard$, 101, 7)))  
O1$ = LTRIM$(RTRIM$(MID$(InCard$, 108, 8)))  
CASE 3  
D1! = FNSINGLE!(MID$(InCard$, 1, 6))+19000000  
D2# = FNDOUBLE#(MID$(InCard$, 7, 5), 5, 0)  
FINISH% = FNINTEGER%(MID$(InCard$, 12, 1))
```

```

                MAKESURE% = FNINTEGER%(MID$(InCard$, 13, 1))
                ENDTIME! = FNSINGLE!(MID$(InCard$, 14, 5))
            END SELECT

            IF (Card = LastCard) THEN
                '*** Insert your program statements here. ***
            END IF
        LOOP
    CLOSE #1
    PRINT "Total cases: "; CaseCount

```

Note: This program, TEST1.BAS, was generated from the questionnaire program called TEST1.QPL, which has been included on the QPL Master Program Disk.

This example is a QBasic program that was generated from a short questionnaire program. You must insert your program lines after the comment, “Insert your program statements here,” in order to make it do something useful, such as printing a list of certain variables.

Options

There is only one option you can use to modify how the CONVERT program creates a QBasic program, in addition to using the Variables menu to specify which variables to convert from strings to numbers. You must set this option on the Options menu before you use this function to generate a QBasic program.

Define Date Components

Dates will normally be translated from a string to a single number in YYMMDD or YYYYMMDD format. You may instead tell the CONVERT program to define each date in three separate variables by checking the “Define date components” option on the Options menu. Instead of converting a date into one single precision number, it will be converted into three integers. It will use the same variable name, but it will append a “Y,” “M,” or “D” to the beginning of each of the three new variable names to indicate which part of the date each contains.

Example

Original Variable Name	Date Components Not Defined	Date Components Defined
D1	D1!	YD1%, MD1%, DD1%

SAS Programs

The SAS for DOS or Windows function generates a SAS program that can read the data files created by the COLLECT and KEYPUNCH programs. You can edit this program with your text editor, or the SAS editor, to further customize how it works. To run the program, you must have a copy of SAS installed on your computer.

You may also use this SAS program on computers with different operating systems. The only parts of the program that may need to be modified are the statements that define the data and library file names. Change these to the file specifications that are appropriate to the operating system your using. The rest of the program should be same for all the platforms that SAS supports.

The CONVERT program will generate a SAS program that

- defines the card and column locations of each variable,
- assigns value labels to multiple choice, check-all-that-apply, and random number questions, and picture formats to numeric and string questions,
- assigns variable labels (the first 40 characters of the question text),
- optimizes the size of the system file, and
- writes frequency and descriptive statistic lists for all the appropriate variables.

Example

```
OPTIONS SOURCE PAGESIZE = 60 LINESIZE = 80;
/*****
/** PROGRAM: A:\TEST1.SAS          DATE: 1-31-1996 **/
/** AUTHOR:                      JOB:                **/
/**                               **/
/** TITLE: QPL Demonstration      **/
/**                               **/
/** PURPOSE:                      **/
/**                               **/
/**                               **/
/**                               **/
/**                               **/
/**                               **/
/*****
TITLE4 'QPL Demonstration';
TITLE5 'Version 4.0';

FILENAME DATAFILE 'A:\TEST1.DAT';

LIBNAME LIBRARY 'A:\';

PROC FORMAT LIBRARY = LIBRARY;

    VALUE _0001_  1 = 'Yes, continue with d'
                2 = 'No, quit the demonstr';

    VALUE _0002_  0 = 'Not checked'
                1 = 'Checked';

    VALUE _0003_  1 = 'Jump to path A'
                2 = 'Jump to path B';
```

Chapter 12
Running the CONVERT Program

```

PICTURE _0004_ LOW-HIGH = '$00,000.00';

PICTURE _0005_ LOW-HIGH = '00,000 days';

VALUE _0006_ 1 = 'Review demonstration'
            2 = 'Save answers'
            3 = 'Erase answers';

VALUE _0007_ 1 = 'Yes, erase these ans'
            2 = 'No, do not erase the';

DATA LIBRARY.TEST1 (LABEL = 'QPL Demonstration');

INFIL DATAFILE N = 3 LINESIZE = 120 MISSEVER;

INPUT #1 @1 VRSN $2.
        @3 INDATE YYMMDD6.
        @9 INTIME 5.0
        @14 M1 1.0
        @15 C101 1.0
        @16 C102 1.0
        @17 C103 1.0
        @18 C104 1.0
        @19 C105 1.0
        @20 C106 1.0
        @21 C107 1.0
        @22 R1 1.0
        @23 N1 4.0
        @27 N2 7.2
        @34 N3 7.2
        @41 N4 7.2
      #2 @1 S1 $100.
        @101 S2 $7.
        @108 O1 $8.
      #3 @1 D1 YYMMDD6.
        @7 D2 5.0
        @12 FINISH 1.0
        @13 MAKESURE 1.0
        @14 ENDTIME 5.0;

LABEL VRSN = '1: Version number of this questionna'
      INDATE = '2: Date questionnaire completed '
      INTIME = '3: Time of day questionnaire complet'
      M1 = '5: Multiple Choice Question Up to 1'
      C101 = '6: First answer'
      C102 = '6: Second answer'
      C103 = '6: Third answer'
      C104 = '6: Fourth answer'
      C105 = '6: Fifth answer'
      C106 = '6: Sixth answer'
      C107 = '6: Other answer'
      R1 = '8: Random Question Random number qu'
      N1 = '11: Number Question Number question'
      N2 = '14: Number Question You got to this'
      N3 = '15: Number Question You got to this'
      N4 = '16: Number Questions You may also u'
      S1 = '17: String Question String question'
      S2 = '18: Formatted String Question Forma'
      O1 = '19: Open-Ended Question This is a l'
      D1 = '20: Date Question Example The date'

```

Chapter 12
Running the CONVERT Program

```

D2          = '21: Compute age in days. This quest'
FINISH     = '23: This is the end of this demonstr'
MAKESURE  = '24: Are you sure you want to erase y'
ENDTIME    = '25: Time questionnaire completed...';

          FORMAT      INDATE      WORDDATE12.
          M1          _0001_.
          C101        _0002_.
          C102        _0002_.
          C103        _0002_.
          C104        _0002_.
          C105        _0002_.
          C106        _0002_.
          C107        _0002_.
          R1          _0003_.
          N2          _0004_.
          N3          _0004_.
          N4          _0004_.
          D1          WORDDATE12.
          D2          _0005_.
          FINISH     _0006_.
          MAKESURE  _0007_.;

          LENGTH      INDATE      4
          INTIME      4
          M1          3
          C101        3
          C102        3
          C103        3
          C104        3
          C105        3
          C106        3
          C107        3
          R1          3
          N1          4
          D1          4
          D2          4
          FINISH     3
          MAKESURE  3
          ENDTIME    4;

PROC FREQ;
  TABLES      VRSN      INDATE;

TITLE6 'Multiple Choice and Random Number Questions';
PROC FREQ;
  TABLES      M1          C101      C102      C103      C104
              C105      C106      C107      R1;

TITLE6 'Number Questions';
PROC FREQ;
  TABLES      N1;

TITLE6 'String Questions';
PROC FREQ;
  TABLES      S2          O1;

TITLE6 'Date Question';
PROC FREQ;
  TABLES      D1          D2;

TITLE6 'Deciding When to Save Records';

```

```
PROC FREQ;  
    TABLES    FINISH    MAKESURE;  
  
TITLE6;  
  
PROC MEANS;  
    VARIABLES N1        N2        N3        N4        D2;  
  
PROC PRINT;  
    VARIABLES S1;  
  
RUN;
```

Note: This program, TEST1.SAS, was generated from the questionnaire program called TEST1.QPL, which has been included on the QPL Master Program Disk.

This is an example of a SAS program that was generated from a short questionnaire program.

Options

There are several options you can use to modify how the CONVERT program writes a SAS program. You must set these options before you generate a SAS program.

Variables Menu

You can shorten the time needed to run your SAS program if you define only the questions you need using the Variables menu. Also, it may be necessary to do this if SAS cannot define your entire questionnaire at one time. This generally is not a problem unless you have more than about 200 questions in your questionnaire. SAS does not have a specific limit on the number of variables that may be defined at the INPUT statement. It keeps adding definitions until it runs out of computer memory. Thus, you may need to experiment to determine how many you can define at one time.

Redefine Long Strings to Short

The CONVERT program normally defines your string questions with the same number of characters that you used to define them in your questionnaire program, up to the SAS maximum of 200 characters. If you have string questions that are longer than this, they will be defined in the SAS program as 200-character strings. This only changes the way that SAS reads your data. The entire string will still be present in your data file, but SAS will read only the first 200 characters of it.

You may instead tell the CONVERT program to truncate the definition of any string questions that are longer than 8 characters by checking “Redefine long strings to short” on the Options menu. (Note: SAS automatically truncates strings when performing some operations. For example, strings are truncated to 16 char-

acters when used in PROC FREQ.) Shortening the definition may allow you to define more variables and make the program run faster.

Define Date Components

The CONVERT program normally defines date questions using the YYMMDD6. or YYMMDD8. input formats. This is usually convenient because you do not need to perform any other transformations on the date before using it to compute the number of days between dates, or before using it in other date functions. It also gives dates a WORDDATE12 output format to make it easier to read the results of a PROC FREQ on a date question.

You may instead tell the CONVERT program to define each part of the date as a separate variable if you check the “Define date components” option on the Options menu. In this case, a date will be defined using three two-character integer fields, and a “Y,” “M,” or “D” will be appended to the beginning of the variable name to indicate what part of the date it is.

Example

Date Components Not Defined				Date Components Defined			
-----				-----			
INPUT #1	@1	VRSN	\$2.	INPUT #1	@1	VRSN	\$2.
	@3	INDATE	YYMMDD6.		@3	YINDATE	2.0
	...				@5	MINDATE	2.0
					@7	DINDATE	2.0
					...		

Numbered Variable Labels

The CONVERT program always writes variable labels for every question that will be defined in your SAS program. It uses the first 40 characters (the maximum SAS will use) of the text of each question for a label. If a the text of a question spans more than one line, the CONVERT program will automatically append lines until it reaches the 40-character maximum.

It is often convenient to have the number of a question (as shown on the COLLECT screen and printed in questionnaire format by the CONVERT program) included in each value label. This makes it easier, for example, to match particular frequency tables to the question as it was originally stated in the questionnaire. You can tell the CONVERT program to include the question number at the beginning of the label by checking the “Numbered variable labels” option on the Options Menu.

Frequency Statistics

Normally, the CONVERT program only writes SAS statements that define your data file. You are expected to add whatever statements you want to use to analyze the data, using your word processor or the SAS editor. You may, however, tell the CONVERT program to write instructions for several procedures that

are commonly used on questionnaire data. The most commonly used procedure is probably PROC FREQ, which will produce tables on each variable that count how many times each answer was chosen by respondents. You can tell the CONVERT program to write PROC FREQ tables by checking the “Frequency statistics” option on the Options menu.

If you check this option, the CONVERT program will list all the questions in a PROC FREQ table that may be appropriate for this type of analysis. These always include multiple choice, check-all-that-apply, random number, and date questions. It will include number questions that are defined as integers (*i.e.*, have no decimal places), and it will include string questions that have a field size of 8 characters or less. You may modify this procedure with your word processor to drop some variables or add SAS options.

The variable lists will also be broken into groups if you have used subtitles to group particular questions. The subtitle you used in your questionnaire program will be used in the SAS program to label the frequency tables on those variables.

Descriptive Statistics

Checking the “Descriptive statistics” option on the Options menu tells the CONVERT program to write a PROC MEANS procedure. This procedure computes simple univariate statistics, including the mean, standard deviation, range, and others. All of the number questions in your questionnaire program will be included in the variable list. You may modify this procedure with your word processor to drop some variables or to add any SAS options.

Long String Lists

Checking the “Long string list” option on the Options menu tells the CONVERT program to write a PROC PRINT procedure. This procedure is useful, for example, when you need to list all the answers to fill-in-the-blank questions. It simply prints all the answers to each question in the variable list. The CONVERT program will include only STRING questions that have a field size greater than 8 characters in the variable list. You may modify this procedure with your word processor to drop or add variables, or to add instructions to tell SAS to only print answers that should be present according to the skip instructions in your questionnaire program.

80- or 132-Column Page

The page size option on the Options Menu does two things: it tells the CONVERT program how wide it may print your SAS program, and it sets the page width for the output of the SAS program when you run it. Use the page size that is convenient for your printer.

SPSS Programs

The CONVERT program generates SPSS-x, SPSS/PC+, and SPSS for Windows programs that can read the data files created by the COLLECT and KEYPUNCH programs. You can edit the program it generates to further customize how it works. To run the program, you must have SPSS installed on your computer. The SPSS-x program and the data file can be uploaded and run on other computers that have SPSS-x software. You must change the data file name specifications in the SPSS-x program, however, to file name specifications that are appropriate to that operating system.

The CONVERT program will generate a SPSS program that

- defines the card and column locations of each variable,
- assigns variable labels (the first 40 characters of the question text),
- assigns value labels to multiple choice, check-all-that-apply, and random number questions, and
- writes frequency and descriptive statistic lists for all the appropriate variables.

Example - SPSS For Windows

```
*****  
** PROGRAM: A:\TEST1.SPS DATE: 1-31-1996 **  
** AUTHOR: JOB: **  
** ** **  
** TITLE: QPL Demonstration **  
** ** **  
** PURPOSE: **  
** ** **  
** ** **  
** ** **  
** ** **  
*****  
TITLE 'QPL Demonstration'.  
SUBTITLE 'Version 4.0'.  
  
DATA LIST FILE = 'A:\TEST1.DAT' FIXED RECORDS = 3  
/ VRSN 1 - 2 (A)  
INDATE 3 - 8  
YINDATE  
MINDATE  
DINDATE 3 - 8  
INTIME 9 - 13  
M1 14  
C1 15 - 21 (A)  
C101  
C102  
C103  
C104  
C105  
C106
```

Chapter 12
Running the CONVERT Program

```

C107      15 - 21
R1        22
N1        23 - 26
N2        27 - 33 (2)
N3        34 - 40 (2)
N4        41 - 47 (2)
/  S1      1 - 100 (A)
  S2      101 - 107 (A)
  O1      108 - 115 (A)
/  D1      1 - 6
  YD1
  MD1
  DD1      1 - 6
  D2      7 - 11
  FINISH      12
  MAKESURE    13
  ENDTIME    14 - 18.

```

VARIABLE LABELS

```

VRSN      '1: Version number of this questionnaire'
/INDATE   '2: Date questionnaire completed '
/YINDATE  'YR: 2: Date questionnaire completed '
/MINDATE  'MO: 2: Date questionnaire completed '
/DINDATE  'DA: 2: Date questionnaire completed '
/INTIME   '3: Time of day questionnaire completed.'
/M1       '5: Multiple Choice Question Up to 19 qu'
/C1       '6: Check-All-That-Apply Question Up to '
/C101     '6: First answer'
/C102     '6: Second answer'
/C103     '6: Third answer'
/C104     '6: Fourth answer'
/C105     '6: Fifth answer'
/C106     '6: Sixth answer'
/C107     '6: Other answer'
/R1       '8: Random Question Random number questi'
/N1       '11: Number Question Number questions ma'
/N2       '12: Number Question You got to this que'
/N3       '13: Number Question You got to this que'
/N4       '14: Number Questions You may also use a'
/S1       '15: String Question String questions ma'
/S2       '16: Formatted String Question Format st'
/O1       '17: Open-Ended Question This is a long,'
/D1       '18: Date Question Example The date ques'
/YD1     'YR: 18: Date Question Example The date'
/MD1     'MO: 18: Date Question Example The date'
/DD1     'DA: 18: Date Question Example The date'
/D2       '19: Compute age in days. This question '
/FINISH   '21: This is the end of this demonstratio'
/MAKESURE '22: Are you sure you want to erase your '
/ENDTIME  '23: Time questionnaire completed... '

```

VALUE LABELS

```

C101
C102
C103
C104
C105
C106
C107      0 'Not checked'
          1 'Checked' /
M1        1 'Yes, continue with d'
          2 'No, quit the demonst' /

```

Chapter 12
Running the CONVERT Program

```
R1          1  'Jump to path A'
           2  'Jump to path B' /
FINISH      1  'Review demonstration'
           2  'Save answers'
           3  'Erase answers' /
MAKESURE    1  'Yes, erase these ans'
           2  'No, do not erase the' .

FREQUENCIES VARIABLES =
  VRSN      YINDATE.

SUBTITLE 'Multiple Choice and Random Number Questions'.
FREQUENCIES VARIABLES =
  M1        C101      C102      C103      C104      C105
  C106      C107      R1.

SUBTITLE 'Number Questions'.
FREQUENCIES VARIABLES =
  N1.

SUBTITLE 'String Questions'.
FREQUENCIES VARIABLES =
  S2        O1.

SUBTITLE 'Date Question'.
FREQUENCIES VARIABLES =
  YD1      D2.

SUBTITLE 'Deciding When to Save Records'.
FREQUENCIES VARIABLES =
  FINISH    MAKESURE.

SUBTITLE 'Version 4.0'.

DESCRIPTIVES
  N1        N2        N3        N4        D2.

LIST VARIABLES =
  S1.

SAVE OUTFILE = 'A:\TEST1.SAV'.
```

Note: This program, TEST1.SPS, was generated from the questionnaire program called TEST1.QPL, which has been included on the QPL Master Program Disk.

Options

There are several options you can use to modify how the CONVERT program writes an SPSS program. You must set these options before you generate an SPSS program.

Variables Menu

You can shorten the time needed to run your SPSS program if you define only the questions you need. You can specify which questions will be defined in the

DATA LIST using the Variables menu. Also, it may be necessary to do this if your questionnaire contains more than 200 variables, the SPSS/PC+ limit.¹³

Redefine Long Strings to Short

The CONVERT program normally defines your string questions with the same number of characters that you used to define them in your questionnaire program. You may, however, run into difficulty using long strings with SPSS/PC+ and because it counts every 8 characters of a string as a separate variable toward the 200-variable maximum.¹⁴ For example, if you defined one string question to have a 20-character field, SPSS/PC+ will count it as 3 variables toward the 200-variable maximum. Thus, it is possible to exceed SPSS/PC+'s variable limitation well before defining 200 questions on the DATA LIST statement.

You may tell the CONVERT program to truncate the definition of any STRING questions that are longer than 8 characters by checking "Redefine long strings to short," on the Options menu. Shortening the definition will allow you to define more variables and may make the program run faster. This option only changes the way that SPSS/PC+ reads your data. The entire string will still be present in your data file, but SPSS/PC+ will read only the first 8 characters.

Define Date Components

The CONVERT program normally defines date questions using a 6-digit number in YYMMDD format or an 8-digit number in YYYYMMDD format. This is may not be convenient, however, if you plan to compute the time between dates because SPSS's YRMODA() function requires that each part of a date be defined as a separate variable. You may tell the CONVERT program to define each part of the date separately if you check the "Define date components" option on the Options menu. In this case, a date will be defined using three two-character fields, and a "Y," "M," or "D" will be appended to the beginning of the variable name to indicate what part of the date it is.

Example

Date Components Not Defined	Date Components Defined
-----	-----
DATA LIST	DATA LIST
FILE = 'C:\TEST1.DAT' FIXED	FILE = 'C:\TEST1.DAT' FIXED
/ VRSN 1 - 2 (A)	/ VRSN 1 - 2 (A)
INDATE 3 - 8	YINDATE
...	MINDATE
	DINDATE 3 - 8
	...

Note: SPSS allows you to define the column locations of a group of variables if they all have the same field size. In this example, two columns will be allocated to each of the three date variables on the right.

¹³SPSS-x and SPSS for Windows allow more variables to be defined but warn that defining a large number of at one time degrades the system performance.

¹⁴This also occurs in SPSS for Windows, but it has a higher variable limit. It suggest that problems will occur if you define more than 4,500 variables at once.

Numbered Variable Labels

The CONVERT program always writes variable labels for every question that will be defined in your SPSS program. It uses the first 40 characters (the maximum SPSS will use) of the text of each question for a label. If the text of a question spans more than one line, the CONVERT program will automatically append lines until it reaches the 40-character maximum.

It is often convenient to have the number of a question (as shown on the COLLECT screen and printed by the CONVERT program) included in each value label. This makes it easier, for example, to match particular frequency tables to the question as it was originally stated in the questionnaire. You can tell the CONVERT program to include the question number at the beginning of the label by checking the “Numbered variable labels” option on the Options Menu.

Frequency Statistics

Normally, the CONVERT program writes only SPSS statements that define your data file. You are expected to add whatever statements you want to use to analyze the data, using your word processor or the SPSS editor. You may, however, tell the CONVERT program to write instructions for several procedures that are commonly used with questionnaire data. The most commonly used procedure, FREQUENCIES, will produce tables on each variable that count how many times each answer was chosen by respondents. You can tell the CONVERT program to write frequency tables by checking the “Frequency statistics” option on the Options menu.

If you check this option, the CONVERT program will list all the variables in a FREQUENCIES statement that may be appropriate for this type of analysis. These always include multiple choice, check-all-that-apply, random number, and date questions. It will include number questions that are defined as integers (*i.e.*, have no decimal places), and it will include string questions that have a field size of 8 characters or less. You may modify this procedure with your word processor to drop some variables or add SPSS options.

The variable lists will also be broken into groups if you have used subtitles to group particular questions. The subtitle you used in your questionnaire program will be used in the SPSS program to label the frequency tables on those variables.

Descriptive Statistics

Checking the “Descriptive statistics” option on the Options menu tells the CONVERT program to write a DESCRIPTIVES procedure for SPSS-x program or a MEANS procedure for SPSS/PC+ or Windows programs. These procedures compute simple univariate statistics, including the mean, standard deviation, range, and others. All of the number questions in your questionnaire program will be included in the variable list. You may modify this procedure with your word processor to drop some variables or to add any SPSS options.

Long String Lists

Checking the “Long string list” option on the Options menu tells the CONVERT program to write a LIST procedure. This procedure is useful, for example, when you need to list all the answers to fill-in-the-blank questions. It simply prints the all answers to each question in the variable list. The CONVERT program will only include STRING questions that have a field size greater than 8 characters in the variable list. You may modify this procedure with your word processor to drop or add variables or to add instructions to tell SPSS to print only answers that should be present according to the skip instructions in your questionnaire program.

80- or 132-Column Page

The page size option on the Options Menu does two things: it tells the CONVERT program how wide it may print your SPSS program, and it sets the page width for the output of the SPSS program when you run it. Use the page size that is convenient for your printer.

Formatted Question List

This function translates your source questionnaire program into a more typical-looking written questionnaire. It numbers the questions, draws boxes for the answers, writes skip instructions, prints the card and column locations for the answers, and divides the document into pages between questions. This document is useful during the development of the instrument, and when training the interviewers, because it shows the question text, answer fields, and skip instructions more clearly than the source questionnaire program does.

The formatted version of your questionnaire will be written to a text file on a disk, and not directly to your printer. After you select this option from the Program menu, a dialogue box will be displayed that asks you to name the file and select any formatting options. The default file name will be the same as your questionnaire program (and be written to the same disk and directory), but the extension will be changed to TXT to indicate that it is an ASCII text file. You may print this file using the File menu Print text file function, or you may print it later using the DOS PRINT command. You also may load it into your word processor for further editing.

Formatted Question List Dialogue Box

Variables Records Options Data Program File Quit

Formatted Question List

File name
A:\TEST1.TXT

Page size
65 Width (10 to 240 columns)
60 Length (10 to 9,999 lines)
5 Left margin (0 to 40 columns)

Options
 Write variable names
 Do not break pages
 Do not write record layout

BEGIN Cancel

A highlight bar indicates which item in the menu you are pointing at. Move the highlight bar up and down using the arrow keys on the right side of your keyboard or by pressing the Enter key.

You may change the page size options by entering a new number. Use the back-space key to erase digits. Press the spacebar to toggle the other options on or off. A check mark indicates when an option is on. These options are described below.

You must press the Enter key to jump from the options to the “Begin” box. Press Enter at this point to start the program, or use the arrow keys to highlight “Cancel” to quit the program. Pressing the up arrow key will take you back to the file name and options. You also may press the Escape key at any time to quit this function and return to the Program menu.

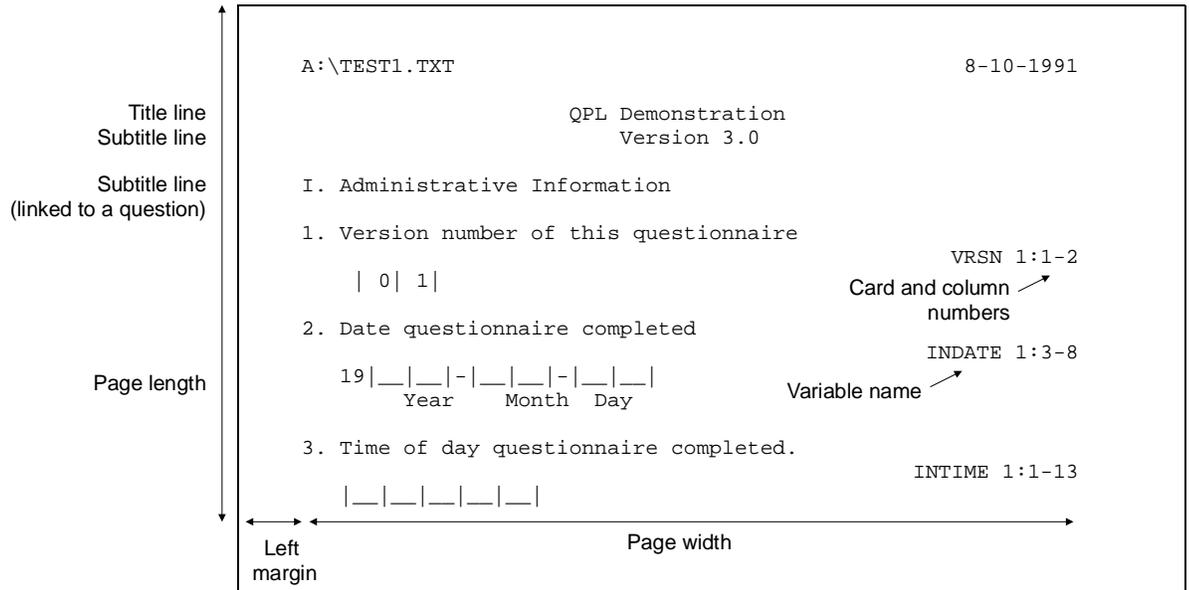
Page Size Options

There are several options that let you determine how the questionnaire will be printed. Using the options in the dialogue box, you can change the default settings for the left margin, and the width and length of the printed page. The diagram on the next page shows how this function measures these settings.

Page Width

The width of the printed page is measured in columns from the end of the left margin. Thus, the sum of the left margin and page width settings will determine the total size of the printed page.

Page Layout



The page width setting determines where right-justified items are printed, such as page numbers, card and column numbers, variable names, and the width of lines it draws for string answers and unconditional (*i.e.*, NEXT) skip commands. It does not, however, affect the length of your question and answer text lines. That is, these lines will be printed exactly as they appear in your source program. A line that exceeds your page width setting will not be “wrapped” to the next line.

Page Length

The page length setting is measured in lines from the top edge of the paper. Two blank lines are always printed at the top of the page. The CONVERT program always checks to see if a question can fit at the bottom of the page. If there is not enough room, it will print a form feed code (*i.e.*, ASCII code 12) to end the page and then begin printing the question at the top of the next page.

Typically, printers will print 66 lines on an 8½-by-11-inch page. Thus, the 60-line default page setting will allow lines to be printed on a page until it reaches line 61. This leaves a minimum margin of ½ inch at the bottom of the page. If you are using legal size paper, which is 14 inches long, you should change the page length setting to 78 lines in order to have a ½-inch margin at the bottom of the page.

Left Margin

The size of the left margin is measured in terms of columns from the left edge of the paper. The default margin setting is five columns. You can set the margin

size from zero to 40 columns. An error message will be displayed on the screen if you enter a number that is out of this range.

Write Variable Names

During the development of your program, it is useful to print a formatted version of your questionnaire that shows the name of each question. This makes it easier to find questions in your source questionnaire document when you need to make changes, and to remember the names of questions when writing SAS or SPSS programs to analyze the data. Each name will be printed to the right of each question, next to the card and column location.

Do Not Break Into Pages

This function will print a question on a new page when it finds that the question will not fit at the bottom of the current page. It will not split a question between two pages. You can, however, tell it to print your questionnaire in one long strip, without any page breaks, by checking the “Do not break into pages” option. It may be useful to do this if you plan to edit the text file with your word processor to add interviewer instructions or other information to the document.

Do Not Write Record Layout

The location of the answer to each question in the data file is printed, in terms of its card and column location, to the right of each question. This information is useful as documentation on how the data file that the COLLECT and KEYPUNCH programs generate is constructed. If you will be using the CONVERT program to generate SPSS or SAS programs, or to translate the data file to dBase or Lotus files, you do not need this information because the CONVERT program automatically takes care of these details.

Titles, Subtitles, and Page Breaks

You can further customize the formatted version of your questionnaire by adding title and subtitles, and by forcing page breaks at particular questions. The title is centered at the top of the first page, then left justified at the top of all succeeding pages. A subtitle line can be added underneath the title line on the first page, and individual subtitle lines can be linked to particular questions so that they can be used to title sections of your questionnaire. Finally, a page break can be used to print a particular question at the top of a page.

TITLE Command

A title must be specified in your source questionnaire program with the TITLE command. Your title must be on one line, and it must be delimited by double quotes. Only one title may be used in your program.

SUBTITLE Command

Subtitles must also be programmed in your source questionnaire program, using the SUBTITLE command. One subtitle may be used with the title. In this case the SUBTITLE command must be used before any questions are programmed. Additional subtitles may be used with particular questions. A subtitle that is linked to a question will always be printed immediately above the question it is linked to,

even if the question must be printed at the top of the next page. Thus, you can give a group of questions a section title by using the SUBTITLE command with the first question in the group. Only one subtitle can be linked with each question.

Example

Source Questionnaire Program	Questionnaire Format
.TITLE = "Title Line"	A:\SAMPLE.TXT
.SUBTITLE = "Subtitle Line"	8/14/89
	Title Line
.QUESTION = Q1, TYPE = DATE	Subtitle Line
.SUBTITLE = "Question Subtitle"	
...	Question Subtitle
...	
...	1. ...
.ANSWER	...
.NEXT	...
	Q1 1:1-6
	_ _ - _ _ - _ _
	Year Month Day

This example shows how the TITLE and SUBTITLE commands are used, and how they would appear in the formatted question list. The title has been set to "Title Line," and its subtitle has been set to "Subtitle Line." A subtitle was also linked to question Q1 by using the SUBTITLE command after its QUESTION command and before its NEXT command.

PAGE Command

You can force a particular question to be printed at the top of a page by using the PAGE command in within the question in your questionnaire program.

Example

Source Questionnaire Program	Questionnaire Format
.TITLE = "Title Line"	Title Line
...	Page 7
...	
	II. Background
.QUESTION = Q37	37. ...
.TYPE = NUMBER	...
.SUBTITLE = "II. Background"	...
.PAGE	
...	Q37 3:14-21
...	_ _ _ _ _ _ _ _ _
...	
.ANSWER = 8	
.NEXT	

In this example, the PAGE command has been used with question Q37. This told the CONVERT program to print this question at the top of a page, regardless of whether it could have fit on the previous page. Here, question Q37 was printed at

the top of page seven. Note that the subtitle that was linked to question Q37 was also printed at the top of page seven.

Cross-Reference List

The CONVERT program can produce a cross-reference listing of your questionnaire program that can be used to track down errors in your skip instructions. The listing shows each question by number, variable name, question type, and card and column location. It also shows (1) all the questions that directly jump to each question and (2) all the questions each question jumps to. This information is useful when the questionnaire logic has become particularly complicated. For example, it will display the phrase “** ERROR **” when there is no way to get to a particular question.

Example

```

A:\TEST1.XRF
8-10-1991 5:14:23 p.m.
Page 1

QPL Demonstration
Version 3.0

Num:Name      Location      Type      Question
=====
1:VRSN        1:1-2         VERSION   Version number of this quest
<---<<      ** START **
>>-->        2:INDATE

2:INDATE      1:3-8         XDATE     Date questionnaire completed
<---<<      1:VRSN
>>-->        3:INTIME

3:INTIME      1:9-13        TIME      Time of day questionnaire co
<---<<      2:INDATE
>>-->        4:TTL

4:TTL         <---<<        VOID      QUESTIONNAIRE PROGRAMMING LA
>>-->        3:INTIME   23:FINISH
5:M1

Multiple Choice and Random Number Questions-----
5:M1         1:14          MULT     Multiple Choice Question Up
<---<<      4:TTL
>>-->        23:FINISH   6:C1

6:C1         1:15-21       CHECK    Check-All-That-Apply Question
<---<<      5:M1
>>-->        7:C2        8:R1

7:C2         <---<<        VOID     Void Question You got to thi
>>-->        6:C1
8:R1         >>-->        8:R1

8:R1         1:22          RANDOM   Random Question Random numbe
<---<<      6:C1        7:C2
>>-->        10:V2       9:V1

```

Chapter 12
Running the CONVERT Program

```

Void Questions-----
      9:V1          VOID      Void Question  The COLLECT pr
      <---<<      8:R1
      >>--->      11:N1

     10:V2          VOID      Void Question  The COLLECT pr
      <---<<      8:R1
      >>--->      11:N1

Number Questions-----
     11:N1          1:23-26    NUMBER     Number Question  Number quest
      <---<<      9:V1          10:V2          12:V3
      13:V4
      >>--->      12:V3          13:V4          15:N3
      14:N2

     12:V3          VOID      You got to this question by
      <---<<      11:N1
      >>--->      11:N1

     13:V4          VOID      You got to this question by
      <---<<      11:N1
      >>--->      11:N1

     14:N2          1:27-33 .2  NUMBER     Number Question  You got to t
      <---<<      11:N1
      >>--->      16:N4

     15:N3          1:34-40 .2  NUMBER     Number Question  You got to t
      <---<<      11:N1
      >>--->      16:N4

     16:N4          1:41-47 .2  NUMBER     Number Questions  You may als
      <---<<      14:N2          15:N3
      >>--->      17:S1

String Questions-----
     17:S1          2:1-100     STRING     String Question  String quest
      <---<<      16:N4
      >>--->      18:S2

     18:S2          2:101-107   STRING     Formatted String Question  Fo
      <---<<      17:S1
      >>--->      19:O1

     19:O1          2:108-115   OPENEND    Open-Ended Question  This is
      <---<<      18:S2
      >>--->      20:D1

Date Question-----
     20:D1          3:1-6       DATE       Date Question Example  The da
      <---<<      19:O1
      >>--->      21:D2          23:FINISH

```

Chapter 12
Running the CONVERT Program

```

21:D2      3:7-11      NUMBER  Compute age in days.  This qu
<---<<   20:D1
>>-->    22:D3

22:D3      <---<<      VOID     Congratulations!  You are #2
>>-->    21:D2
          23:FINISH

```

Deciding When to Save Records-----

```

23:FINISH  3:12      MULT     This is the end of this demo
<---<<   ** ESCAPE **   5:M1      20:D1
          22:D3      24:MAKESURE
>>-->    4:TTL      25:ENDTIME  24:MAKE

24:MAKESURE 3:13      MULT     Are you sure you want to eras
<---<<   23:FINISH
>>-->    23:FINISH  25:ENDTIME

25:ENDTIME  3:14-18 TIME     Time questionnaire completed
<---<<   23:FINISH  24:MAKESURE
>>-->    ** END **

```

=====

```

C1.....6
C2.....7
D1.....20
D2.....21
D3.....22
ENDTIME...25
FINISH....23
INDATE....2
INTIME....3
M1.....5
MAKESURE...24
N1.....11
N2.....14
N3.....15
N4.....16
O1.....19
R1.....8
S1.....17
S2.....18
TTL.....4
V1.....9
V2.....10
V3.....12
V4.....13
VRSN.....1

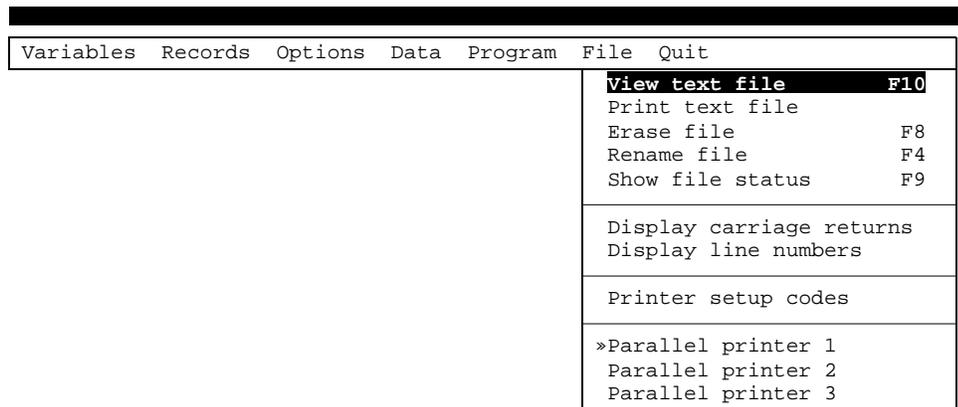
```

Notes: The "<---<<" symbol indicates which questions immediately precede a question, and ">>-->," indicate which questions immediately follow a question. This report, TEST1.XRF, was generated from the questionnaire program called TEST1.QPL, which has been included on the QPL Master Program Disk.

File Menu

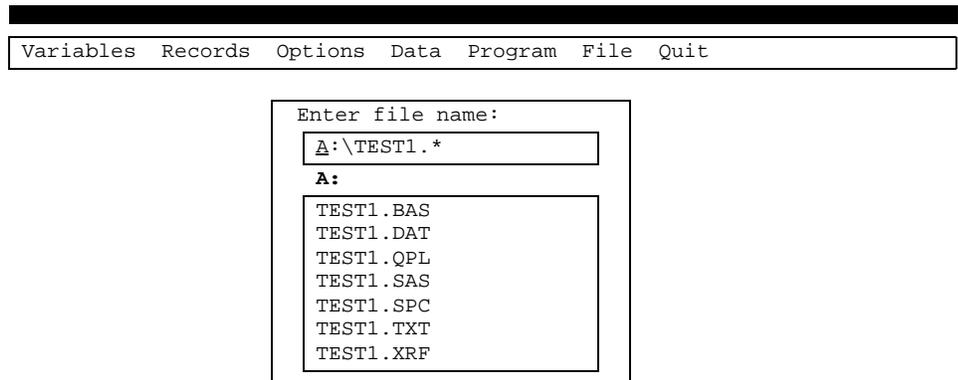
The File menu contains several functions that allow you to manipulate your disk files. It has functions that you can use to view or print text files (such as SPSS and SAS programs) and erase, rename, and show when a file was created and how large it is.

File Menu



When you select one of these functions, a dialogue box will be displayed in the middle of the screen that asks you to enter the name of the file you want to use. It also shows all the files that match the name of your current questionnaire program (on the same disk drive and in the same directory). There are two ways to enter a file name. You can edit the file name in the upper box to an exact name (*i.e.*, a file name that does not have the DOS “?” or “*” wild card characters). Press the Enter key when you have finished editing the name, or press the Es-cape key to abort the function and return to the File menu.

File Directory Dialogue Box



You also may select one of the files that are listed in the lower box if you press the Enter key without changing the default file name. In this case, a highlight bar

will appear on the first file in the list. You may then select a file by using the up and down arrow keys to move the highlight bar to the file you want and then pressing the Enter key to select it. Press the Escape key to back up to the file specification line.

If the file you want is not listed in the lower box, you can edit the file name (using the DOS wild card characters, “?” and “*”) to specify a match by a different file name. You can also change the disk drive and directory specifications. Press the Enter key to get a new list of matching file names. Then press the Enter key again to jump to the lower box to select the one you want.

View Text File

You can use the “View text file” function to look at files that contain only ASCII text characters, such as your source QPL program or the BASIC, SAS, or SPSS programs that are generated by the CONVERT program. It is often useful to take a quick peek at the SAS or SPSS program that you generate to make sure you included all the variables and other options that you intended to use. The CONVERT program will load up to 1,000 lines of your document.

After you have selected a text file, the CONVERT program will display it in a window in the middle of the screen. If the file is too long to fit in this window, you may use the up and down arrow keys to scroll the file up and down one line at a time, or the Page Up and Page Down keys to scroll by one window page at a time.

Display Carriage Return

There are two options you may use with the view function. You can display the carriage returns (called “[HRT]” by WordPerfect) on the screen if you check the “Display carriage return” option on the File menu screen. Carriage return codes will be displayed as left-pointing triangles.

Display Line Numbers

You may also display the number of each line of the document if you check the “Display line numbers” option on the File menu screen. The line numbers will be written on the left side of each line.

Print Text File

The “Print text file” function can be used to print any of the program or report files that the CONVERT program generates, or any other line-oriented text file. Use the file directory dialogue box to select the file you want to print, such as an SPSS or SAS program file. After you have selected a file, another dialogue box will be displayed that shows how much of the file has been printed. You can stop printing at any time by pressing the Escape key. Your printer may not stop immediately, however, if it uses a printer buffer to store incoming characters.

Printer Setup Codes

You should not have to do anything special to make the COLLECT program work with your printer. Since it sends only standard ASCII character and carriage control codes to the printer, it should work with any printer that supports ASCII codes. You may, however, use this function to change the character font and pitch, line size, or other features your printer offers.

When you select the “Printer setup codes” function, a dialogue box will be displayed in the middle of the screen that asks for setup and reset control code strings. Most printers are programmed using control codes or escape sequences. Consult your printer’s manual for the specific codes it uses. The CONVERT program will send the setup string you enter to your printer before it starts printing the report to initialize your printer. After the report is finished, it will send the reset string to return the printer to its default settings.

Type control code numbers using a backslash and then the number of the code (as a three-digit decimal number, using leading zeros if necessary). For example, to send an escape code (27) to the printer, you should type “\027” in your setup string.

You also may use ASCII characters to represent control code numbers. For example, Epson printers are programmed using escape sequences. These sequences begin with the escape code, followed by one or more character codes that specify what to change. To tell an Epson printer reset all of its options to the factory defaults, you could enter “\027@” or “\027\064.” Since the @ character has an ASCII value of 64, both of these strings send the same codes to the printer.

Generally, using the ASCII characters makes the code strings shorter to write and a little easier to read. Since upper and lowercase letters have different ASCII values, however, you must be careful to type characters exactly as shown in your printer manual. Do not type any spaces between the codes and letters unless your printer manual specifically says to do so.

Printer Initialization Files

The printer setup and reset strings you write can be saved in a file so that they may be automatically reloaded the next time you start the CONVERT program. The KEYPUNCH program also uses printer initialization files. After you have finished entering both the setup and reset strings, another dialogue box will be displayed that asks if you want to save the strings in a file. If you answer “Yes,” your strings will be written to a file called “QPLPRINT.INT” on the same disk and in the same directory as your QPL program. The current page size settings (*i.e.*, the number of columns and lines per page) also will be saved in this file.

Several printer initialization files have been included on the QPL Master Program Disk. They are listed in the table below.

Printer Initialization Files on QPL Master Program Disk

EPSON.INI	Sets print for 132 columns and 80 lines per page for Epson FX and LQ series printers.
HPLJIICL.INI	Sets print for 176 columns and 60 lines per page in landscape mode for Hewlett-Packard LaserJet Series II printers and compatibles.
HPLJIICP.INI	Sets print for 132 columns and 80 lines per page in portrait mode for Hewlett-Packard LaserJet Series II printers and compatibles.
IBMPRO.INI	Sets print for 132 columns and 80 lines per page for IBM ProPrinters and compatibles.
OKIDATA.INI	Sets print for 132 columns and 80 lines per page for Okidata Microline 82A and 93 dot matrix printers.

QPL printer initialization files are simple ASCII text files. You may view or edit these files using the DOS EDLIN program or your word processor. Lines that begin with an asterisk are comments. The first noncomment line must be the setup string, the second line must be the reset string, and the third line must contain the number of columns and the number of lines per page.

Example — EPSON.INI

```
*
* EPSON.INI      - Various Epson printers that are using
*                the Epson mode including:
*                FX, FX-80, and FX-286
*                LQ-850, LQ-950, and LQ-1050,
*                17 pitch type
*                8 lines per inch
*
* Set KEYPUNCH page size options to:
*
*                220 column page (wide paper), or
*                132 column page (letter-size paper)
*                80 lines per page
*
\027@\027\015\027\048
\027@
132 80
```

To use one of the printer initialization files, first copy the file you want to use to the disk and directory where your QPL application program is located, and then change the name to “QPLPRINT.INI” (*i.e.*, the default file name). Then, when you start the CONVERT program, this file will be automatically loaded.

You may instead tell the CONVERT program the name and location of a particular initialization file by starting it with the /I command line option.

Example — /I Option

```
C:\> CONVERT C:\JOB1\MYJOB.QPL /IC:\QPL\EPSON.INI
```

In this example, command line options are used to tell the CONVERT program to load a questionnaire program and a printer initialization file. The questionnaire program is called “MYJOB.QPL,” which is located on the C: drive and in the “\JOB1” directory. The /I option tells it to load the printer initialization file called “EPSON.INI,” which is located in the “\QPL” directory on the C: drive.

Parallel Printer Port

The last three options on the File menu tell the CONVERT program which printer port to use. The first parallel port (LPT1: or PRN:) is used by default. If you want to use another port, move the highlight bar to the one you want to use and then press the Enter key. The check mark indicates which port is currently selected.

The CONVERT program cannot be configured to print through a serial port.

Erase File

The “Erase file” function will delete a file on your disk. After you have selected a file, using the directory dialogue box, another dialogue box will be displayed on the screen that asks you to confirm the name of the file you specified to be deleted. Select “Yes” to erase the file listed, or “Cancel” to return to the File menu without erasing the file. Use the left and right arrow keys to highlight your choice, and then press the Enter key to start the function. You may instead press the Escape key to abort the erase function and return to the File menu without erasing the file.

Rename File

The “Rename file” function will give an existing file a new name. After you have selected a file using the directory dialogue box, another dialogue box will be displayed that asks for the new file name. You may either edit the name of the file you selected (which is already filled in) or give it an entirely new name. You also may change the directory name, but you may not change the disk drive. Press the Enter key when you finished editing the file name, or press the Escape key to return to the File menu.

After you have entered a new name, another dialogue box will be displayed that asks you to confirm the name change. Select “Yes” to rename the file listed or “Cancel” to return to the File menu without renaming the file. Use the left and right arrow keys to highlight your choice, and then press the Enter key to start the function. You may instead press the Escape key to abort the rename function and return to the File menu without renaming the file.

Installing the QPL Programs

This appendix provides a sample setup for a hard disk-based computer. This information is intended for the person who writes questionnaire programs. An interviewer disk setup for the COLLECT program is described in chapter 9, a network setup for the NCOLLECT program is described in chapter 10, and a disk setup for the KEYPUNCH program is described in chapter 11.

Master Program Disk Files

QPL System Programs	Demonstration Files

COLLECT EXE	CLEAN BAT
COMPILE EXE	C1 DAT
CONVERT EXE	C1BAK DAT
KEYPUNCH EXE	C2 DAT
NCOLLECT EXE	C2BAK DAT
MONITOR EXE	MEWA2 DAT
	C1 DOC
Utility Program	C2 DOC
-----	MEWA DOC
SQUASH EXE	MEWA2 DOC
	Q1 DOC
	Q2 DOC
	TEST1 DOC
	DEMO EXE
	LOOK EXE
	MAKEDEMO PRG
	C1 QPL
	C2 QPL
	MEWA QPL
	MEWA2 QPL
	Q1 QPL
	Q2 QPL
	TEST1 QPL
	INTRO TXT
	TEST1 TXT
	Foreign Language Files

	MESSAGE DOC
	SPANISH DOC
	Printer Initialization Files

	EPSON INI
	HPLJIICL INI
	HPLJIICP INI
	IBMPRO INI
	OKIDATA INI
	Product Information Files

	FILELIST TXT
	INSTALL TXT
	README TXT

Note: Installing all of the files requires 1.6 megabytes of hard disk space. Installing only the system programs requires 1.3 megabytes.

You should not use the Master Program Disk to run the QPL system programs, including the demonstration. It should be used only to copy the QPL programs to your floppy or hard disk. Keep the original Master Program Disk in a safe place as a backup.

Developer's Hard Disk Setup

All of the QPL system programs should be copied to a new directory on your hard disk drive, such as "C:\QPL."

Hard Disk Setup

```
C:\QPL                C:\job
-----
QPL System            Application and
Programs              Data
-----
COMPILE.EXE          filename.ext
COLLECT.EXE          filename.QPL
CONVERT.EXE          filename.DAT
KEYPUNCH.EXE         ...
NCOLLECT.EXE
MONITOR.EXE
```

Note: The word "job" refers to the name of the directory where your questionnaire program, data, and other files are located. The demonstration files also could be copied to the C:\QPL directory.

You also should set a pathway to the C:\QPL directory. This allows you to execute the QPL programs from the directory where your source questionnaire program is located.

Example

```
C:\> PATH C:\;C:\DOS;C:\QPL
C:\> CD \job
C:\job> COMPILE TEST1.DOC
```

This example shows how to set a path to the C:\QPL directory. You could instead include the path command in your AUTOEXEC.BAT file so that the pathway to the C:\QPL directory will be set when you boot up the computer.

Installation Steps

The QPL programs and demonstration files have been put on the QPL Master Program Disk in a single compressed file. A minimum 1.6 megabytes of available disk space is required to install these files.

Follow the steps below to install these files in a directory on your computer's hard disk.

Installing the QPL Software From the Master Program Disk

1. Make a directory for these files on your computer's hard disk (such as \QPL, for example).
C:\ MD QPL
2. Make this your default directory.
C:\ CD QPL
3. Put the QPL Master Program disk in drive B:. (Note: You may instead use your computer's A: drive. Be sure to change the B: to A: when extracting the files in step 4.)
4. Extract the QPL programs and files from the Master Program Disk:
C:\QPL B:QPL40

All of the QPL files have now been copied into your \QPL directory. You may now type DEMO at the DOS command line to run a short demonstration of the software and to review the changes to the software. The information about the changes is also included in the DOS text file on the Master Program disk called README.TXT.

Installation on a Network

You may use these same installation steps to put the QPL system programs on a network disk drive if you will be the only person using the software. Just change the disk drive letter in the above example to one of your network drives.

See chapter 10 for instructions on how to install this software with your questionnaire on a network for use in a CATI survey.

Demonstration Files

The demonstration files can be used to get a quick overview of the QPL system of programs. They provide a short description of the system and let you run several short questionnaire programs with the COLLECT, KEYPUNCH, and CONVERT programs.

To start the demonstration, make the disk drive and directory where the files are located the default, then type DEMO and press the Enter key. Press the Escape key to quit the demonstration.

Appendix I
Installing the QPL Programs

A batch file, called CLEAN.BAT, may be used to erase any files that may have been created when running the demonstration.

Changes From Previous Versions

Important!

Version 4.0 of the QPL system of programs for computer-aided telephone interviewing (CATI) and data entry have been revised to add features that support networking and to update some of the data conversions.

Existing questionnaire programs must be recompiled before they can be used with the new QPL system programs. This is necessary because the structure of the QPL file has been changed to accommodate the new commands. The new QPL system programs will display an error message if you try to load a Version 3.x or 2.x program.

In Version 4.0, two new programs have been added to the system, called NCOLLECT and MONITOR, which allow you to install your questionnaire on a network and have two or more interviewers simultaneously use the same questionnaire and data files. This arrangement can greatly simplify the administration of a CATI since you do not have to setup disks for each interviewer. It is a further benefit if your survey uses a control file, because any of the interviewers can select any outstanding case and because the NCOLLECT program automatically prevents interviewers from selecting the same case.

You do not need to do anything special in your questionnaire program to take advantage of these features. In fact, you can use the same compiled questionnaire file on a network (with NCOLLECT) and on a laptop PC (with COLLECT). The data files that will be created are the same and may be merged together for analysis.

The MONITOR program is a utility that lets you watch how the CATI is progressing. You may use it to see, from your network workstation, what case each interviewer is currently working on, what cases have been completed, and how many cases have been put in the data file.

See chapter 10 for more information on how to set up a network-based CATI survey.

The other QPL system programs have also been updated.

COLLECT and KEYPUNCH Programs

The following changes have been made to the COLLECT and KEYPUNCH programs.

- A message is displayed that tells interviewer to press the spacebar to check answers in a check-all-that-apply question.
- A message is displayed that tells an interviewer to press the Esc (escape) key to exit an open-end question.

- A date stamp is only put in an OPENEND response file the first time an interviewer answers the question.
- Command line options have been added to set a user name and to pass a number to the questionnaire program

KEYPUNCH Program

The following changes have been made to the KEYPUNCH program.

- The sort routine has been made much faster by using extended or expanded memory.
- Printer LPT ports can now be selected from the Print menu.

CONVERT Program

The following changes have been made to the CONVERT program.

- Tab delimited files can be created.
- A documentation box is no longer put at the top of Lotus 1-2-3 files (making them easier to import to other packages).
- askSam import files can be created.
- Awk programs that can create summary statistics on the questionnaire data files can be created. (If you have a Thompson Awk compiler, you can use these Awk programs to create a mini-analysis package that you can give to others who are using your questionnaire but do not have access to SAS or SPSS.)
- The Basic program has been revised to generate only QBasic code, which is compatible with Visual Basic.
- SPSS for Windows programs can be created.

The QPL language has also been expanded to add several keywords that make it easier to control a questionnaire's operation from the command line when starting the program and to take advantage of network capabilities.

NEW Keyword

The NEW keyword is true if an interviewer is working on a case for the first time. It is false if the case was reloaded from the data file. This keyword is used to control actions that should only occur once when the case is first attempted, such as setting the case identification number with the COUNT keyword (see below).

COUNT Keyword

The COUNT keyword returns a unique case identification number. It can be used to automatically number cases. This is especially useful if your questionnaire uses open-ended questions. Since these questions create a text file that uses the case identification number as a name, the COUNT keyword can be used to guarantee that interviewers always use a unique number.

OPTION Keyword

The OPTION keyword's value is set by the /O# option at the DOS command line when the COLLECT, NCOLLECT, or KEYPUNCH programs are started. This lets you

pass a number from the command line to your questionnaire. You could use it, for example, to control what questions are asked by using it to compute a value for a question, which is later used by skip instructions.

WORKSTATION Subargument

The WORKSTATION subargument lets you automatically keep track of who conducted each interview. It copies an interviewer's network name as an answer to a STRING question. The NCOLLECT program will get the name from the network (without displaying this question to the interviewer) and then move to the next question. An interviewer's network name can also be set with the /W command line option when starting this program.

Changes From Version 2.0 to 3.0

The following discussion has been retained for QPL users who may be upgrading to Version 4.0 from Version 2.0. Version 4.0 retains all of the features that were added to Version 3.0:

- several new types of questions: check-all-that-apply, 8-digit date, and long open-ended questions (see CHECK, LDATE, and OPENEND),
- new types of skip instructions (see SKIP and SKIPIF),
- date and math functions (see TOJUL(), ROUND(), LOG(), etc.),
- computed variables (see COMPUTE and COMPUTEIF),
- improved control file support (see SAVECONTROLIF, RELOADIF, PAUSE, and COPY),
- improved screen color and other settings (see FOREGROUND, BACKGROUND, etc.), and
- programmable foreign language error messages.

These new types of questions and commands make it easier to program complex questionnaires and to customize an interview for a particular respondent. They also make it easier for the interviewer to use the programs, for example, by enabling them to reload incomplete interviews and restart them where they left off (when using control files). You also may change the screen colors for particular questions to visually define different parts of your questionnaire for the interviewer, such as an error message.

Although the COMPILE, COLLECT, KEYPUNCH, and CONVERT programs look similar to their Version 2.0 counterparts, they also have been improved. Most of the changes have been made to accommodate the additions to the language, such as the new types of questions. In addition to these changes, the KEYPUNCH program, for example, now can handle questionnaires with up to 1,056 questions. It also can sort interview records, delete an interview record, and create a text-

based data file that can be used with text-retrieval and hypertext software called askSam.

The COLLECT program also has been improved. It now allows negative numbers to be entered in a NUMBER question, which also has a pop-up calculator that the interviewer can use to compute answers. The new open-ended question allows interviewers to write several pages or more for an answer to a single question. This question has basic word-processing features such as word-wrapping, block copying, block deleting, and word searching that make it easy for the interviewer edit a long response.

The CONVERT program now has a printing function that can be used to print your formatted questionnaire documents or SAS and SPSS program listings. This function also can be programmed to select particular fonts, type sizes, and margin settings when printing your document. Printer settings for several popular printers have been included on the Master Program Disk.

The PUBLISH program has been merged into the CONVERT program as a function in the Program menu.

IF and GOTO Commands

In Version 2.0, both IF and GOTO commands could be used within the same MULT or RANDOM questions. All of the GOTO commands, however, would be evaluated before any IF commands regardless of the order in which they were typed. These commands, and the new SKIP and SKIPIF commands, are now evaluated in the order in which they were typed in a question.

Example

Source program	Questionnaire Format
.QUESTION = Q1, TYPE = MULT What is your favorite color? .ANSWER Red Orange Yellow Green Blue Indigo Violet Other... .IF (M1 = 8) Q2 .GOTO = Q3 .NEXT = Q4	1. What is your favorite color? (CHECK ONLY ONE ANSWER) __ 1. Red __ 2. Orange __ 3. Yellow __ 4. Green __ 5. Blue __ 6. Indigo __ 7. Violet __ 8. Other... (GO TO #3) IF (#1 = 8) GO TO #2 SKIP TO QUESTION 4 =====

If you used this question in a Version 2.0 program and the interviewer selected the last answer, “Other,” the COLLECT program would skip to question Q3 because it checks all of the GOTOs first. Under Version 3.0, the COLLECT program will skip to question Q2 because the IF command was used before the GOTO command.

FOREGROUND, BACKGROUND, HIGHLIGHT, BORDER, CLOCK, UPPERCASE, and BEEPER Commands

In Version 2.0, the commands that tell the COLLECT program how to display questions, FOREGROUND, BACKGROUND, HIGHLIGHT, BORDER, CLOCK, UPPERCASE, and BEEPER, could be used only once in a program, and they would affect all of the questions regardless of where in the program they were used (except for CLOCK, which could be used only within questions).

These commands may now be used to change the settings for individual questions, ranges of questions, or all the questions in your program.

If you use one of these commands outside of a question, it will change the setting for all of the subsequent questions, or until the command is used again. (Thus, if you have used these commands before any questions in your existing programs, you will not notice any difference when you recompile and run them with the COLLECT program.) If you use one of these commands within a question, the setting will be changed for only that question; the previous setting will be restored at the next question.¹

SAVEIF Command

In Version 2.0, the SAVEIF command could be used to specify when an interview record should be saved and when a control file record should be updated. In this version, the SAVEIF command determines only when an interview record should be saved. A new command, SAVECONTROLIF, has been added to specify when the control file record should be updated. A control file record will be updated every time an interview is completed if the SAVECONTROLIF command is omitted, regardless of whether the interview record will be saved.

CARD Command

A error in the CARD command in Version 2.0 caused answer fields to be wrapped to the next data card one column too soon. For example, if the default card setting of 80 columns was used, the COMPILE program would wrap data fields to the next card after the 79th column was used instead of the 80th column, unless a STRING question was set to exactly 80 columns.

¹See chapter 9 for examples of how these commands may be used.

Because the CARD command in Version 1.0 worked correctly, the record layouts of the same program compiled under each version could be different, and thus you could not use a data file that was created with Version 1.0 with Version 2.0.²

This error has been fixed in Version 3.0. If possible, each data card will be filled to the 80th column (assuming the default CARD size). Now the record layouts of data files created with Version 3.0 will match Version 1.0, but not Version 2.0. You can tell the Version 3.0 COMPILE program to wrap fields at the same point as the Version 2.0 COMPILE program by using the new command called FIX.

Example

```
.FIX = OFF
```

Setting the FIX command to OFF, as shown in this example, tells the COMPILE program to use a Version 2.0 record layout. The default setting is ON (use a Version 3.0 layout).

NUMBER Question Type

In Version 2.0, only positive numbers could be entered as a response to a NUMBER question. Version 3.0 allows the interviewer to enter negative numbers. If your existing programs depend on zero being the lowest value an interviewer may enter, you should use the LOW command to explicitly tell the COLLECT and KEYPUNCH programs not to allow negative responses.

Example

```
.QUESTION = Q1, TYPE = NUMBER = "$###,###"  
...  
.ANSWER  
.LOW = 0  
.NEXT
```

In this example, the lowest amount the interviewer can enter is \$0, because the LOW command was set to zero. If the LOW command was not used, the lowest amount that could be entered would be \$-99,999. (Note that the minus sign uses one of the digits in the field; it always floats immediately in front of the number.) The highest amount that could be entered is \$999,999.

²Generally, however, this problem could be overcome by setting the CARD command to 81 instead of 80.

Changes From Version 1.0 to 3.0

If you are currently using QPL Version 1.0, you also must consider the following syntax changes before recompiling your existing questionnaire programs. As with the changes between Versions 2.0 and 3.0 and, most of the language has not changed, but there are some differences you may have to accommodate.

The utility programs have also been enhanced. The PUBLISH, TONEW, TOFREE, TOSAS, TOSPSS, and CROSSREF programs have been combined into one program called CONVERT. This new program also can translate QPL data files directly into Lotus 1-2-3 and Symphony spreadsheets, and dBase III+ and IV DBF files. It allows you to select variables and records to keep or drop during the translation of the data files.

FORMAT Command

The FORMAT command is no longer used. QPL programs now use only multiple card, fixed-format records. The CONVERT program can be used to translate data from this format to Lotus WKS, WK1 and WR1 files; standard delimited PRN files; and dBase DBF files. The COMPILE program will give you a warning message if you use this command in your program, but it will still create a QPL file.

MISSING Command

The MISSING command is no longer used. A blank space is always used to fill a field that is off the skip path or marked as not known by the interviewer. You can set missing values to -1 when translating a data file to comma delimited, Lotus, or dBase formats using the CONVERT program. The COMPILE program will give you a warning message if you use this command in your program, but it will still create a QPL file.

LOW and HIGH Commands

The LOW and HIGH commands have been enhanced to allow complex expressions to be used to set range limitations. Numbers must now be specified with a decimal point, if necessary.

LANGUAGE Command

The LANGUAGE command is no longer used. Instead, error messages may be translated by setting a message number equal to the translated message on a command line in your program.³ Numbers, times, and dates still automatically conform to local formatting conventions according to the country defined in your CONFIG.SYS file.

Installation Procedures

Setting up the QPL system programs for use by interviewers or by the questionnaire program developer has been simplified. The BRUN30.EXE and QPLLIB.EXE programs are no longer needed to run the system programs, and it is no longer necessary to set a LIB environment variable in DOS.

³See chapter 7.

Language Summary

This appendix summarizes the QPL syntax and lists all of the command and key words, and equation operators and functions. Refer to the chapters at the beginning of this manual for examples of how they are used in a questionnaire program.

Program Lines

QPL programs must be saved as ASCII text files. Each line in the program may have one of five different purposes: program commands, comments, blank lines, and question and answer text lines. Each program line should end with carriage return and line feed codes (called “[HRT]” in WordPerfect). Do not use any special formatting features your word processor may provide, such as bolding or underlining.

Example

```

Comment line  → ** Series of questions on instruction techniques.
Blank line   →
Command line → .QUESTION = Q34, TYPE = MULT
Question text lines → Will instruction be presented through class
                    → lectures and discussions, OR will it be
                    → presented on an individual, self-paced
                    → basis?
Command line  → .ANSWER
Answer text lines → Lecture and discussion
                    → Self-paced
                    → Both lecture and self-paced
                    → Other
Command lines  → .GOTO = Q35
                    → .NEXT = Q36

```

Note: A period must be used in the first column of a command line and an asterisk in the first column of comment line. Text lines may begin with any other characters.

Commands and Keywords

The following tables summarize all the QPL commands and keywords, and how they are used. The underlined portion of these words indicate how they may be abbreviated in your questionnaire program. For example, the QUESTION command may also be written as “QU” in your program. Commands and keywords also may be written using upper or lowercase letters. The square brackets show when command arguments are optional, and the words in lowercase letters show what type of arguments are used.

Program Flow Commands

ANSWER [= name to copy] or
[= field size]

Define end of question text and beginning of answer text. It can also be used to specify the length of string and numeric fields, define the default multiple choice answer, and copy an answer format and field size from a previous question.

<code>COMPUTE (expression)</code>	Enter the result of the expression as a response to a multiple choice, check-all-that-apply, number, date, or time question. The question will not be displayed by the COLLECT program on the screen. The COLLECT and KEYPUNCH programs will evaluate the expression and use it as a response as the interviewer passes through the question.
<code>COMPUTEIF (test expression) (expression)</code>	If the test expression evaluates to a true value, enter the result of the expression as a response to a multiple choice, check-all-that-apply, number, date, or time question. The question will not be displayed by the COLLECT program on the screen. Any number of COMPUTEIF commands may be used within a question. The COLLECT and KEYPUNCH programs will evaluate COMPUTEIF test expressions until one evaluates to a true value. A COMPUTE command should be used to set the question to a default value when all the COMPUTEIF test expressions are false.
<code>ESCAPE = name</code>	Specify the question to jump to when the interviewer presses the Escape key. If no question is defined, the COLLECT and KEYPUNCH programs will jump to the last question in the questionnaire.
<code>GOTO = name</code>	Specify a conditional skip based on the response to a multiple choice, check-all-that-apply, or random question. If a GOTO command is not linked to a particular answer, then the COLLECT program will skip to the question that is specified by an IF statement or the NEXT command. GOTO commands take precedence over IF commands, regardless of how they are ordered in your program.
<code>IF (expression) variable</code>	Specify a conditional skip based on the response to a multiple choice, check-all-that-apply, random, date, or time question. It must be used within a question. The IF command can be used within any type of question, and any number of IF statements can be used. If more than one IF statement is used, the COLLECT program will skip to the question that is specified by the first true statement. Subsequent IF statements and the NEXT command are ignored.
<code>NEXT [= name]</code>	Define end of answer list and question set. It may also be used to specify an unconditional skip to another question. The NEXT command may be superseded by a skip that has been specified by a GOTO or IF command.
<code>QUESTION = name</code>	Define start of a question and assign it a variable name.
<code>SAVEIF (expression)</code>	Specify when the COLLECT and KEYPUNCH programs should save an interview record in the data file. An expression can be used to base this action on the response to any number of answers to the questionnaire.
<code>SKIP (target expression)</code>	Jump to a question according to its number instead of its name. The target expression will be evaluated to determine the number of the question. This skip instruction is generally used to jump to one of a consecutive series of questions. You

can determine the number of the first question in the series by prefacing its name with an exclamation mark in the expression. Then the COLLECT and KEYPUNCH programs will use the question's number as a value in an expression instead of its current response. (ex. SKIP (!Q1 + Q2 - 1)

SKIPIF (test expression)
(target expression)

If the test expression is true, then jump to a question according to its number instead of its name. The target expression will be evaluated to determine the number of the question. This skip instruction is generally used to jump to one of a consecutive series of questions. Any number of SKIPIF commands may be used within a question. Precede a question's name with an exclamation mark to tell the COLLECT and KEYPUNCH programs to use its number as a value instead of its current response.

Question Types

TYPE = type

Define type of question. Must be used after a QUESTION command and before the ANSWER command. See following types of questions.

TYPE = CHECK

Check-all-that-apply question. An answer list must be used after the ANSWER command. Each line of the list is a separate answer. An interviewer may check any, all, or none of the answers. Use the GOTO command or the IF command with the @ operator to write skip instructions that are based on the answers that were selected. The names of CHECK questions should not be more than six characters long.

TYPE = DATE

Six-digit date question. This question presumes that the 20th century is being used, that is, the dates will fall between 1900 and 1999. If a default date is not used on the answer text line, the current date will be initially displayed to the interviewer. The names of DATE questions should not be more than seven characters long.

TYPE = LDATE

Eight-digit date question. This question allows the interviewer to enter dates between years 1 and 9999. If a default date is not used on the answer text line, the current date will be initially displayed to the interviewer. The names of LDATE questions should not be more than seven characters long.

TYPE = MULT

Multiple choice question. An answer list must be used after the ANSWER command. Each line of the list is a separate answer. The answer list may be as long as will fit on the screen when combined with the question text and the three lines of the question box.

TYPE = NUMBER [= decimal places] or [= "format"]

Number question. No decimal places need to be specified if your number is an integer; otherwise, you must specify how many decimal places you want to use.

The total length of the field must be defined with the ANSWER command, or by specifying a default answer after the ANSWER command. You may instead use a format string to create a template of how you want your number to appear to the interviewer. A “#” character defines where the interviewer may type a number (ex. “\$#,###.##”). A format string also automatically sets the field length and number of decimal places.

TYPE = OPENEND

Open-ended question. To accommodate the large variation in the amount of text an interviewer may enter, the responses to this question will be stored in separate files. These files will always have an OTX file name extension. The field in the main data file will contain only the name of the file that holds the response. This file name is constructed from a unique identification number and the name of the question. You must use the IDENTIFY command to tell the COMPILE program which question will be used to hold this number. The name of the question, which cannot be more than four characters long, will be appended to this number when the file name is created. The default edit buffer is set to 4,000 characters (about two double-spaced pages). The buffer size can be set to 1,000 to 32,000 characters using the ANSWER command.

TYPE = RANDOM

Random number question. This question type is programmed just like a multiple choice question. It is not, however, displayed on the screen by the COLLECT program. It will randomly select one of the list of answers and then immediately skip to the question that is specified by the GOTO, IF, or NEXT commands.

TYPE = STRING [= "format" or
WORKSTATION]

Short answer question. Short answer questions may be up to 32,000 characters long, although few other analytical programs can handle strings of this length. Check the analytical software you plan to use for limitations on string lengths. The length of the field may be specified using the ANSWER command to set the field size or by typing a default answer after the ANSWER command. Format strings can be used to create a template of how you want the answer field to appear to the interviewer. The “#” character defines where a number may be typed, “@” defines where a letter may be typed, and an “*” defines where a number or letter may be typed. Only the characters that the interviewer enters are saved in the data file. The format string also sets the field length. The WORKSTATION subargument instead tells the COLLECT or KEYPUNCH programs to copy the interviewer’s network name as an answer (or the name set at the DOS command line with the /W option). The field size is automatically set to 45 characters (the maximum size of a Novell NetWare user name). You may shorten the field size by using the ANSWER command. A STRING question is not displayed to the interviewer if the WORKSTATION subargument is used.

TYPE = TIME

Automatically answered time question. A default answer does not need to be entered after the ANSWER command. The time is set when the interviewer first

passes through the question. It is not changed if the interviewer backs up and goes through the question again.

TYPE = VERSION

Questionnaire program version number. Always writes the default answer specified after the ANSWER command to each record.

TYPE = VOID

Void question. This question does save any data in the interview record. It may be used to give the interviewer supplementary information or an error message. The default answer line is displayed on the screen by the COLLECT program and usually should just be a short message to the interviewer such as, "Press ENTER to continue..."

TYPE = XDATE

Automatically answered 6-digit date question. A default answer does not need to be entered after the ANSWER command.

Data Format Commands

CARD = number

Specify the maximum number of characters that may be put on one line of a record. Default is 80 characters. Minimum is 1 character, maximum is 32,000 characters. Note: SPSS/PC+ card limit is 1,032 characters.

FIX = ON or OFF

Turn the correction to the CARD command on or off. This command tells the COMPILE program to generate a Version 2.0 record layout by ignoring a correction to the CARD command. If you set FIX to OFF, it will use the Version 2.0 layout. If you set it to ON, the default setting, it will use the corrected CARD setting (which also matches data files created with Version 1.0).

FORMAT

This command is no longer used. The CONVERT program should be used to translate the QPL fixed-format records into comma delimited PRN files; Lotus WKS, WK1, or WR1 files; or dBase DBF files.

MISSING

This command is no longer used. The COLLECT and KEYPUNCH programs will always use a blank space (ASCII code 32) to fill a field that is off the skip path, or a field that has been marked as not known using the question mark key.

NEWCARD

Begin the answer to this question at the beginning of the next line of the multiple card, fixed-format record. This command must be used within QUESTION/NEXT set.

Control File Commands Used in a Control File Program

HEADER = "phrase... [control
name]"

Define how the list of respondents will be displayed on the Control List screen.

IDENTIFY = name

Define which question will be used to hold a unique case identification number. This case number may be from one to four digits long, and it must be stored in a NUMBER question. This information is used by the RELOADIF command to identify which incomplete interview record should be reloaded and restarted by the COLLECT or KEYPUNCH programs.

RELOADIF (expression)

Reload an incomplete interview if the interviewer selects it from the Control List screen. The test expression evaluates information in the control file data base to determine if an incomplete interview record exists. If it does, it will be loaded into the COLLECT or KEYPUNCH programs. The IDENTIFY command must be used with this command to tell these programs which question holds a unique case identification number. This information is used to find the correct case in the interview data file. The PAUSE command may be used in the questionnaire program to tell these programs where the interview should be restarted.

SHOWIF (expression)

Specify when a record from a control file will be displayed for the interviewer to choose from. For example, you can specify an expression that only lists respondents that have not been called.

Control File Commands Used in a Questionnaire Program

CONTROL = filename.QPL

Specify a QPL program to be used to control the list of respondents that are called. The control program must exist before the questionnaire program can be compiled.

COPY = name

Copy the response from one question to another. The name of the question to be copied must be specified, and the COPY command must be used within the question that is receiving the response. The COLLECT and KEYPUNCH programs allow the interviewer to edit the copied response. The response, however, will be copied only the first time the interviewer sees a question. Both questions must have the same type and field size.

IDENTIFY = name

Define which question will be used to hold a unique case identification number. This case number may be from one to four digits long, and it must be stored in a NUMBER question. This information is used by the RELOADIF command to iden-

tify which incomplete interview record should be reloaded and restarted by the COLLECT or KEYPUNCH programs. It is also used by OPENEND questions to name OTX files.

INCREMENT = control name

Increase the value of the current answer of a variable in the control file by one. This command can be used to count how many times an interviewer tried to reach a particular respondent. The argument must be a number question in the control file. This variable is only incremented when the control data file is updated.

GET = control name

Copy a question and its current answer from a control file. This command must be used outside of a QUESTION/NEXT set.

PAUSE = name

Define which question will be used to hold the number of the question that COLLECT displayed on the screen before the interviewer pressed the Escape key. This information is used by the RELOADIF command to restart an incomplete interview at the point where it was interrupted. This question must be a NUMBER. The answer to this question may be set using a COMPUTE command and the ESCAPE expression keyword (ex. COMPUTE (ESCAPE)).

PUT = control name

Copy the answer to this question to a question in the control program. This command must be used within a QUESTION/NEXT set. The variable name used in the argument must be a variable in the control file, and both questions must have the same type and field size.

SAVECONTROLIF (expression)

Specify whether a control file record should be updated after an interview is completed. The control file record will be updated if the expression is true. The control file record will always be updated if the SAVECONTROLIF is not used.

COMPILE Commands

DISPLAY = ON or OFF

Display the source code on the screen as it is being processed by the COMPILE program.

INCLUDE = filename

Specify an additional source file for your questionnaire program. The COMPILE program will combine any number of source questionnaire files into one QPL file. Included files may also be nested, that is, an included file may include another file.

VERBOSE = ON or OFF

Display all error messages during COMPILE. Normally, only errors that were not caused by other errors are displayed by COMPILE. To see all of the error messages, set this command to on. Default is off.

COLLECT, NCOLLECT, and KEYPUNCH Commands

`number = "message"`

Change a default English error message to another language. All of the error and other messages the COLLECT program displays, and many of the messages the KEYPUNCH program displays, may be changed. A particular message may be changed by setting its number to a new message. A list of all the messages and their numbers have been included in a file called MESSAGE.DOC on the QPL Master Program Disk. The Spanish translation of some of these messages has been included in a file called SPANISH.DOC. Typically, you would create on program that contains your translated messages and then merge it into your questionnaire programs using the INCLUDE command. This allows you to keep your translations separate from your various questionnaire projects. The translations will be included in your compiled QPL file.

`BACKGROUND = color`

Specify the color that will be used for the background color by the COLLECT, KEYPUNCH, and CONVERT programs. (See list of colors below.)

Screen Colors

<u>BLACK</u>	<u>GRAY</u>
<u>BLUE</u>	<u>LIGHTBLUE</u>
<u>GREEN</u>	<u>LIGHTGREEN</u>
<u>CYAN</u>	<u>LIGHTCYAN</u>
<u>RED</u>	<u>LIGHTRED</u>
<u>MAGENTA</u>	<u>LIGHTMAGENTA</u>
<u>BROWN</u>	<u>YELLOW</u>
<u>WHITE</u>	<u>BRIGHTWHITE</u>

`BACKUP = filename`

Specify back up data file name for the COLLECT program to use. A back up file is created only when a filename is specified. The KEYPUNCH program will does not create back up files.

`BEEPER = ON or OFF`

Turn error beep off or on. Default is on.

`BORDER = color`

Specify the color that will be used for the border lines in the COLLECT, KEYPUNCH, and CONVERT programs. (See list of colors above.)

`CLOCK = ON or OFF`

Show the elapsed time that a question has been displayed on the screen by the COLLECT program. This command is not used by the KEYPUNCH program.

`FOOTER = "phrase... [name]"`

Specify a line of text that will be displayed at the bottom of the COLLECT screen. It is not used by the KEYPUNCH program. The text line may include responses to other questions in the survey. For example, it could be programmed to display

	the name of the current respondent. The current answer to a question may be displayed by typing its variable name, delimited by square brackets, in the text line. Only one FOOTER command may be used in a questionnaire program.
<code>FOREGROUND = color</code>	Specify the color that will be used for the characters on the COLLECT, KEYPUNCH, and CONVERT programs. (See list of colors above.)
<code>HEADER = "phrase... [name]"</code>	Specify a line of text that will be displayed at the top of the COLLECT screen. It is not used by the KEYPUNCH program. The text line may include responses to other questions in the survey. For example, it could be programmed to display the name of the current respondent. The current answer to a question may be displayed by typing its variable name, delimited by square brackets, in the text line. Only one HEADER command may be used in a questionnaire program. This command must be used in a control file program. In this case, it defines how the list of respondents will be displayed on the Control List screen.
<code>HIGH = (expression) or "string"</code>	Specify the highest value that may be entered for this question. The COLLECT and KEYPUNCH programs will not allow the interviewer to move to another question unless an answer is entered that is equal to this value or lower. The value is not checked if the interviewer has entered question mark (<i>i.e.</i> , don't know) response.
<code>HIGHLIGHT = color</code>	Specify the color that will be used for error and other messages by the COLLECT program. The KEYPUNCH program uses this color to indicate which questions are not on the skip path or automatically answered. The CONVERT program uses this color to list questions that will be dropped when the data file is translated to other formats. (See list of colors above.)
<code>LOW = (expression) or "string"</code>	Specify the lowest value that may be entered for this question. The COLLECT and KEYPUNCH programs will not allow the interviewer to move to another question unless an answer is entered that is equal to this value or higher. The value is not checked if the interviewer has entered question mark (<i>i.e.</i> , don't know) response.
<code>PASSWORD = number</code>	Specify a password number the interviewer must give in order to start the COLLECT and KEYPUNCH programs. If no password is used, then no password is needed to start these program. A password must be a number between 1 and 32,000.
<code>UPPERCASE = ON or OFF</code>	Use only uppercase, or upper and lowercase, letters for answers to string questions. Default is only uppercase letters.

CONVERT Commands

<u>PAGE</u>	This command is used when generating a formatted questionnaire document. It tells the CONVERT program to print the question in which it used at the top of the next page.
<u>SUBTITLE</u> = "phrase"	<p>This command tells the CONVERT program to insert a subtitle in a formatted questionnaire document. It can be used two ways. First, it may be used outside a question to define a subtitle that will be printed under the title on the first page.</p> <p>Second, it may be used within a question to link a subtitle to a particular question. These subtitles will be printed immediately above the question it was used in. Subtitles are also used by the CONVERT program when it generates SAS and SPSS programs, and by the KEYPUNCH program when it prints formatted interview records. A subtitle may be up to 64 characters long.</p>
<u>TITLE</u> = "phrase"	Specify the questionnaire title. The title will be printed at the top of each page of the formatted questionnaire document. It will also be used to title SAS and SPSS programs. The KEYPUNCH program also uses it to title formatted interview records. A title may up to 70 characters long.

QPL Expressions

Many of the QPL commands accept expressions for their arguments. These include the IF, SHOWIF, SAVEIF, LOW, HIGH, and other commands. QPL expressions are similar to those you may have used in other programming languages, but with the following two exceptions:

- An expression must be enclosed in parentheses. This tells the COMPILE program where the expression begins and ends on a command line, and separates the expression from other commands that may be on the same command line.
- An expression must be on a single line. The COMPILE program will give you an error message if the expression is split across two or more lines.

QPL expressions are evaluated from left to right, according to the hierarchy shown in the following table.

QPL Hierarchy of Operations

- 1. NOT, functions
- 2. ^ or ** (exponentiation)
- 3. *, /, @
- 4. +, -
- 5. >, <, >= or =>, <= or =<
- 6. =, <> or ><
- 7. AND
- 8. OR

As this table shows, for example, multiplication and division are performed before addition or subtraction, and both are done before any tests of equality are made. This hierarchy can be overridden by using parentheses to group operations. There is no limit on how many parentheses may be used.

In addition to numbers and variables, there are several QPL keywords that can be used in expressions.

QPL Expression Keywords

COUNT	A unique case identification number if the interviewer is working on a new case or zero if the case was reloaded from the data file (range 1 to 32,000)
DATE	The current system date in YYYYMMDD format.
ESCAPE	The number of the question that was displayed before the Esc key was pressed
FALSE	A number whose value is "false" (<i>i.e.</i> , zero).
MISSING or ?	A number that has a missing value.
NEW	True if the interviewer is working on a new case or false if the case was reloaded from the data file
OPTION	Number entered at the DOS command line using the /O# option when the program was started.
TRUE	A number whose value is "true" (<i>i.e.</i> , any non-zero number).
PASSWORD	The current interviewer's password number.
RECORD	The current record number.
TIME	The current system time as seconds from midnight.

The TIME and DATE keywords should be used only with the SHOWIF, COMPUTE, or COMPUTEIF commands. The RECORD keyword is available only when writing expressions interactively using various menu functions in the KEYPUNCH and CONVERT programs.

The MISSING keyword may be used to test whether the answer to a numeric question has been set to "Don't know" by the interviewer. To test for missing values, you must explicitly compare a variable name to the missing keyword (ex. IF (Q7 = MISSING) Q14). Numeric questions that have been set to a missing value will be defined to have a value of zero in all other operations.

The RECORD keyword may only be used when searching for records using the KEYPUNCH or CONVERT programs. It may not be used in any expressions in your questionnaire program.

Date Functions

TOJUL()	Return Julian date from Gregorian date
FROMJUL()	Return Gregorian date from Julian date
JDAYOFWEEK()	Return a code that indicates the day of week from a Julian date
JFHOLIDAY()	Return a code that indicates a weekend or U.S. federal holiday from a Julian date
JYEAR()	Return year from Julian date
JMONTH()	Return month from Julian date
JDAY()	Return day from Julian date
GYEAR()	Return year from Gregorian date
GMONTH()	Return month from Gregorian date
GDAY()	Return day from Gregorian date

Notes: Arguments used with functions that begin with the letter J, and the FROMJUL() function, must be Julian dates. Arguments used with functions that begin with the letter G, and the TOJUL() function, must be Gregorian dates written in YYYYMMDD format. Answers to DATE and XDATE questions are automatically promoted to YYYYMMDD format when used in expressions.

JDAYOFWEEK() Return Codes

- 0 Invalid date
 - 1 Sunday
 - 2 Monday
 - 3 Tuesday
 - 4 Wednesday
 - 5 Thursday
 - 6 Friday
 - 7 Saturday
-

JFHOLIDAY() Return Codes

- 0 Weekday (non-holiday)
- 1 Weekend
- 2 New Year's Eve
- 3 Martin Luther King Day
- 4 President's Day
- 5 Memorial Day
- 6 Independence Day
- 7 Labor Day
- 8 Columbus Day
- 9 Veterans Day
- 10 Thanksgiving
- 11 Christmas

Math Functions

ABS()	Return absolute value of a number
SGN()	Return sign of a number (1 if positive or zero, 0 if negative)
ROUND()	Return integer that has nearest value to a real number
FLOOR()	Return largest integer that is less than or equal to a real number
CEIL()	Return smallest integer that is greater than or equal to a real number
MOD()	Return the decimal portion of a real number
LOG()	Return the natural log of a number
EXP()	Return the exponential of a number (the inverse of the natural log)
LOG10()	Return the base-10 log of a number
SQR()	Return the square root of a number

All of the QPL functions will return zero if they are given invalid arguments. You should check the value of any question that could cause an error before using it as an argument for one of these functions.

Warning

The variables you use in IF statements should refer only to questions that the interviewer has answered. You cannot be assured of a particular value for a question if it is not on the path that the interviewer has taken. Questions on other paths, or questions that are farther down the path, may not contain the default answers you specified because the interviewer may have gone down that path entering data, then backed up and proceeded down a new path.

Command Line Options

All of the QPL programs have options that can be set when starting them from the DOS command line, including the name of the file you want to process. If you do not specify a file when starting a program, a “Startup Menu” will be displayed. This menu asks for the name of the file you want to use and lets you select any options. You may select additional options from the command line. Generally, these options are used less often. All of the command line options are listed below.

Square brackets are used below to indicate optional information. Lowercase words indicate what type of argument is required, and a “#” symbol is used to indicate when a number is required. Do not type a space between an option letter and any numbers, filenames, or letters that it may require. You must, however, type at least one space between each option you use.

COMPILE

```
COMPILE [sourcefile.ext...] [/B] [/Ccontrolfile.QPL] [/D]
        [/Errorlog.ERR] [/H] [Oobjectfile.QPL] [/M] [/N] [/P#]
```

```
/B      Set screen colors to black and white
/Cfile  Set control file name
/D      Display source files on the screen
/Efile  Rename error log file
/H      Display command line options
/Ofile  Rename QPL output file
/M      Show all error messages (default is first error per question)
/N      Do not pause at end of compilation
/P#     Set password number (range 1 to 32767)
```

Note: Up to 10 source files may be specified at the command line. Multiple files will be merged into one QPL file. The name of the first file will be used to name the QPL file.

COLLECT

```
COLLECT [filename[.QPL]] [/B] [/E] [/Fbackupfile.ext]
        [/H] [/O#] [/P#] [/Q] [/R] [/T] [/U] [/Wname]
```

```
/B      Set screen colors to black and white
/E      Set error beep off
/Ffile  Specify backup data file name
/H      Display command line options
/O#     Set OPTION keyword number
/P#     Enter password number (range 1 to 32767)
/Q      Do not display question number
/R      Do not restart interview
/T      Do not display program title
/U      Set uppercase control off
/Wname  Set workstation name
```

Appendix IV Command Line Options

NCOLLECT

```
NCOLLECT filename[.QPL] [/B] [/C] [/D#] [/E] [/H] [/O#]
        [/P#] [/Q] [/R] [/T] [/U] [/Wname] [/X#] [/Z]

/B      Set screen colors to black and white
/C      Use first available control file case
/D#     Maximum number control cases displayed (default 50)
/E      Set error beep off
/H      Display command line options
/O#     Set OPTION keyword number
/P#     Enter password number (range 1 to 32767)
/Q      Do not display question number line
/R      Do not restart interview
/T      Do not show program title
/U      Set uppercase control off
/Wname  Set workstation name
/X#     Set shared file access retry time (default 10 seconds)
/Z      Do not ask network for workstation name
```

Note: A file name must be entered at the command line.

MONITOR

```
MONITOR filename[.QPL] [/B] [/D#] [/H] [/T]

/B      Set screen colors to black and white
/D#     Maximum number cases displayed (default 100)
/H      Display command line options
/T      Do not show program title
```

KEYPUNCH

```
KEYPUNCH [filename[.QPL]] [/B] [/F] [/H] [/Ifilename] [/L#]
        [/O#] [/P#] [/Wname]

/B      Set screen colors to black and white
/F      Print reports to a file
/H      Display command line options
/Ifile  Printer initialization file (default is QPLPRINT.INI)
/L#     Set parallel printer port (default is 1)
/O#     Set OPTION keyword number
/P#     Enter password number (range 1 to 32767)
/Wname  Set work station name
```

CONVERT

```
CONVERT [filename[.QPL]] [/B] [/H] [/Ifilename] [/L#]

/B      Set screen colors to black and white
/H      Display command line options
/Ifile  Use alternate printer initialization file
/L#     Set parallel printer port number (default LPT1:)
```

Appendix IV Command Line Options

DEMO

```
DEMO      [filename.ext] [/B] [/H] [/F#] [/S#]

filename.ext  Menu script and actions (default is MAKEDEMO.PRG)
/B           Set screen colors to black and white
/H           Display this help screen
/F#          Set character color by number
/S#          Set screen color by number

           0 Black    4 Red
           1 Blue     5 Magenta
           2 Green    6 Brown
           3 Cyan     7 White
```

DEMO is a general purpose demonstration program. You may set it up to run any DOS programs by modifying the script file. The script file is used to tell the DEMO program how to list each of your applications in the menu and the command line needed to start each application. By default, it looks for a script file called MAKEDEMO.PRG. The following table shows how the script file is structured.

DEMO Script Format

```
Title Line 1
Title Line 2
Box Title
Action 1 Label for menu
Action 1 DOS command line (for color monitor)
Action 1 DOS command line (for monochrome monitor)
Action 2 Label for menu
Action 2 DOS command line (for color monitor)
Action 2 DOS command line (for monochrome monitor)
etc.
```

Note: A maximum of 16 action items may be used.

LOOK

```
LOOK      filename.ext [/B] [/N]

/B         Set screen colors to black and white
/N         Show line numbers
```

Note: LOOK displays line-oriented text files on the screen. It will load up to 1,000 lines of the file and let you scroll backward and forward through the file without altering it.

SQUASH reduces an existing file's size to zero bytes without changing its network file attributes. This is useful alternative to deleting a network data file that has been given shared and read-write attributes (because the new data file that NCOLLECT will create will not have the shared attribute).

SQUASH

```
SQUASH    filename.ext [filename.ext...]
```

File Name Extensions

The following tables list the file name extensions that are used by the QPL system programs. Generally, all the files that QPL programs create will have the same name as your original questionnaire program and be written to the same disk drive and directory, but a new three-letter extension will be used to indicate what type of file it is.

COMPILE

QPL	Compiled questionnaire program. This questionnaire is file used by all other QPL system programs. A formatted questionnaire may be generated and printed using the CONVERT program.
ERR	Error log file. Lists all errors that may have occurred when processing your source questionnaire program. It may be printed using the DOS PRINT command or loaded into your word processor.

COLLECT

DAT	Fixed format data file. It contains interview records that may be edited or printed using the KEYPUNCH program or translated to other formats using the CONVERT program.
LOGIC.ERR	Record dump on logic error. This file is created if a logic error occurred after the interviewer attempted to save an interview record. It is a text file that reports (1) the response to each question, (2) which questions were on the skip path, and (3) when and where the logic error occurred. It may be printed using the DOS PRINT command or loaded into your word processor. The data in this file will not be added to the main data file. You must use this information to rekey the data after correcting the logic problem.
OTX	Response to an open-ended question. The name of each file is constructed from a unique case identification number and the name of a question. You should use the KEYPUNCH program to view or print open-ended responses with the responses to other questions. You may also use the CONVERT program to merge all the responses from one or more questions into an ASCII text file that can be loaded into your word processor. You should <u>not</u> edit these files with your word processor. They contain formatting codes that are unique to QPL.
CNT	A text file that contains the last COUNT keyword number used. You can set a new starting number by editing this file. The count number must be the very first item in the file.

NCOLLECT

DAT	Fixed format data file. (See description above.)
LOGIC.ERR	Record dump on logic error. (See description above.)
OTX	Response to an open-ended question. (See description above.)
CAT	Catalog file. A binary file that keeps track of what interviews are currently being conducted. A file size of zero indicates that no interviews are being conducted. You may use the MONITOR program to view the catalog information.
JRN	Journal file. A binary file that records all of the attempted interviews, whether or not an interview record was saved, since the system was started. You may use the MONITOR program to view the journal information. This file is not needed to run the system. It is provided only for project management purposes. It may be safely deleted at any time.
CNT	A text file that contains the last COUNT keyword number used. (See description above.)

KEYPUNCH

DAT	Fixed format data file. It contains interview records that may be translated to other formats using the CONVERT program.
DA!	Copy of the fixed format data file before the last sort. This file is created when you use the Save menu Sort function. It contains the data in its original, unsorted, order.
DEL	Deleted record file. This file contains fixed format records that were removed from the main data file using the Save menu Delete record function.
INI	Printer initialization file. Several files with this extension have been included on the QPL Master Program disk that contain printer setup codes for different printers. The file called QPLPRINT.INI is created when you use the File menu Printer setup codes function to create or edit setup and reset strings. The KEYPUNCH program will automatically load this file if it is on the same disk and in the same directory as your compiled questionnaire program. (See also the /I command line option.)
KTX	Formatted interview records. Generally used to create an askSam text-based data file.
LOGIC.ERR	Record dump on logic error. (See description above.)
OTX	Response to an open-ended question. (See description above.)
CNT	A text file that contains the last COUNT keyword number used. (See description above.)

CONVERT

ATX	askSam import data file.
AWK	Thompson Awk program.
BAS	QBasic interpreter program.
CTX	Merged open-ended responses put into ASCII text file format. This file may be loaded into your word processor for further editing.
DAT	Fixed format data file. It may be edited using the KEYPUNCH program.
DBF	dBase III+ or IV data file.
INI	Printer initialization file. (See description above.)
PRN	Comma or tab delimited data file. Also called a free format data file. This is a generic type of data file that may be used with many statistical and general purpose programming software packages.
SAS	SAS/PC program.
SPC	SPSS/PC+ program.
SPS	SPSS for Windows program.
SPX	SPSS-x program.
TXT	Formatted questionnaire document. It is used for reviewing the original questionnaire program. It may be printed using the CONVERT program File menu Print function.
WKS	Lotus version 1A spreadsheet file.
WK1	Lotus version 2.01 spreadsheet file.
WR1	Lotus Symphony spreadsheet file.
XRF	Cross-reference listing of questions and skip instructions and record layout of main data file. It may be printed using the CONVERT program File menu Print function.

Index



- * 33, 248, 255
 - ?
 - See also MISSING keyword
 - ? keyword 255
 - @ operator 73, 93
 - Example 22
 - @ sign 25, 33, 125, 248, 255
 - askSam record delimiter 179, 194
-
- A**
- ABS function 79, 82, 257
 - AND operator 255
 - ANSWER command 7, 245
 - Copy answer list 13
 - Copy answer text 35
 - Copy number answer line 27
 - Copy short answer format 35
 - Copying number answer 29
 - Answer lines
 - Random number question 62
 - Answers
 - Text lines 7-8
 - Append records 196
 - Arithmetic operators 73
 - ASCII
 - Answer range 37
 - Fixed format data 88
 - Source program 5
 - askSam 193
 - Open-end questions 39
 - Asterisk 33, 248, 255
 - Arithmetic operator 73
 - See Comment lines
 - DOS file name wild card 229
 - GET questions 107
 - ATX file extension 193, 263
 - AUTOEXEC.BAT 235
 - AWK file extension 263
-
- B**
- Back up data file 175-176
 - BACKGROUND command 144, 173, 242, 252
 - Open-ended question 175
 - BACKUP command 139, 252
 - Backup data file 138
 - Bad file number 161
 - BAS file extension 263
 - BASIC
 - Define date components 208
 - Maximum card size 89
 - Select variables 186
 - View program listing 229
 - Batch file
 - Multiple compiles 130
 - Starting the COLLECT program 123
 - BEEPER command 137, 242, 252
 - Binary arithmetic 93
 - Blank lines 7
 - Blank space
 - See Command lines
 - Leading blanks in fixed format file 94
 - Missing value code 91, 244
 - Number question 25
 - Short answer question 94
 - Blank spaces 33
 - Bolding 5
 - BORDER command 144, 173, 242, 252
 - BRUN30.EXE 244
-
- C**
- C1.BAK 234
 - C1.DAT 234
 - C1.DOC 234
 - C1.QPL 234
 - C2.DAT 234
 - C2.DOC 234
 - C2.QPL 234
 - C2BAK.DAT 234
 - CARD command 35, 89, 242, 249
 - Caret 255
 - Carriage return code 5, 91
 - Display when viewing text file 229
 - Case has not been checked out 162
 - Case is already checked out 162
 - Case of letters 37
 - CAT file extension 152, 262
 - CATI 1
 - See COLLECT program
 - See NCOLLECT program
 - CEIL function 79, 83, 257
 - Check-all-that-apply question 18
 - Field size 92
 - CLEAN.BAT 234, 237
 - CLOCK command 145, 242, 252
 - See also Current time question
 - CNT file extension 76, 152, 261-262
 - COLLECT program 135
 - Backup data file 135, 138
 - Change from previous version 238
 - Check disk space 148
 - Check time and date 148
 - Color keywords 143
 - Description 2
 - Display elapsed time 145
 - Don't display program title 140
 - Don't display question number 140
 - Don't restart interview 140
 - DOS command line options 137, 258
 - Error beep 137
 - File name extensions 261
 - Headers and footers 146
 - Interviewer's disk setup 147
 - Password 140
 - Quitting 149
 - Screen display 142
 - Set workstation name 141
 - Setting screen colors 143
 - Uppercase option 141
 - COLLECT.EXE 234
 - Color graphics adapter 128, 137, 143, 154, 165
 - Color keywords 143, 252
 - Color of screen 173
 - COLLECT program 137, 143
 - COMPILE program 128
 - CONVERT program 182
 - KEYPUNCH program 171
 - MONITOR program 165
 - NCOLLECT program 154
 - Comma
 - See Command lines
 - Comma delimited data files 198
 - Comma,European number format 120
 - Command line options 258
 - Command lines 6, 245
 - Blank space 6
 - Comma 6
 - Letter case 6
 - Multiple commands 6
 - Command summary 245
 - Comment lines 7
 - COMPILE program 124
 - Description 1
 - Display command line options 132
 - Do not pause after 131
 - DOS command line options 125, 258
 - Errors 132
 - File name extensions 261
 - Rename QPL file 131
 - Set password 131
 - Stop compile 134
 - COMPILE.EXE 234
 - COMPUTE command
 - Multiple choice questions 14
 - Number 29
 - COMPUTEIF command
 - Multiple choice questions 17
 - Number 29
 - Computer memory 5
 - Conditional skips
 - See GOTO and IF commands
 - CONFIG.SYS
 - Country 120, 244
 - CONFIG.SYS file (DOS) 119, 127
 - CONTROL command 106, 130, 250
 - Control file 99
 - Automatically use first free case 154
 - COMPILE program 125
 - CONTROL command 106
 - GET command 106
 - HEADER command 103
 - Importing schedule data 118
 - Limit display of cases 154
 - PUT command 108
 - Set at compile time 130
 - SHOWIF command 102

CONVERT program 180, 244
 BASIC program generator 206
 Change from previous version 239
 Copy records 195
 Cross-reference list 225
 Data dialogue boxes 192
 Description 3
 DOS command line options 182, 259
 Erase file 232
 File dialogue box 228
 File name extensions 263
 Free format data files 198
 Function keys 185
 Keys used for menus 185
 Lotus 201
 Menus 183-184
 Password 183
 Printer control codes 230
 Printer initialization files 230
 Printer port 232
 Printing 229
 Quitting 184
 Rename file 232
 SAS program generator 208
 Screen colors 182
 Select records 188
 Select variables 186
 Select variables dialogue box keys 188
 SPSS program generator 215
 Translate data 196
 View text files 229
 CONVERT.EXE 234
 COPY command 117, 250
 COPY command (DOS) 100
 Copy data to control file 108
 Copying data from control file 106
 Copying records 188
 COUNT keyword 74-75, 239, 255
 CNT file 152
 COUNTRY command (DOS) 119
 Cross-reference list 3, 225
 CROSSREF.EXE
 CONVERT program 244
 CTX file extension 263
 Current date question 51
 Default answer 52
 Field size 52, 96
 Promotion to YYYYMMDD 78, 96, 256
 Text lines 51
 XDATE keyword 51
 Current time question 53
 Elapsed time between questions 54
 Internal format 54
 TIME keyword 53
 When time is recorded 53

D

DA! file extension 262
 DAT file extension 138, 152, 261-263

Data analysis 180
 Data collection instrument
 See Formatted question list
 Data editing 169
 See KEYPUNCH program
 Data files
 Appending interview records 135
 Backup 138
 Created by COLLECT and KEYPUNCH
 88
 dBase 3
 Field width 7
 Free format 3
 Lotus 3
 QPL fixed format 88
 DATA LIST command (SPSS) 218
 Data quality 11
 Data translation
 See CONVERT program
 Date
 Defined in BASIC program 208
 Defined in free field file 199
 DATE command (DOS) 148, 175
 Date functions 78, 256
 DATE keyword 74, 247, 255
 When not to use 71
 Date of interview
 See Current date question
 Date question 43
 See also Current date question
 DATE keyword 44
 Default answer 45
 European format 119
 Field size 44, 96
 Limited by current date 52
 Limiting answer range 46
 Promotion to YYYYMMDD 78, 96, 256
 Use instead of DATE keyword 71
 Date, Check for correct on start up 148
 Dates
 Defined in SAS program 213
 Lotus 202
 SPSS 218
 dBase 195
 Character fields 35
 COPY TO SDF 118
 Date field 96
 European formatted numbers 120
 Field size 195
 Record number field 195
 Redefine character field size 195
 Reset don't know response 195
 Select fields 186
 Select records 188
 USE command 195
 DBF file extension 195, 263
 Decimal point
 European number format 120
 Fixed format file 94
 Use in LOW and HIGH commands 244

Default answer
 Current date question 52
 See Multiple choice question
 Short answer question 35
 Version number question 59
 DEL file extension 262
 Delete file 232
 Delete records 196
 Deleting interview records 85
 Deleting records 188
 DEMO program
 DOS command line options 260
 DEMO.EXE 234, 236
 Demonstration files 234, 236
 DESCRIPTIVES command (SPSS-x) 219
 Disk setup
 Hard disk 235
 Disk, check free space 148, 176
 Display cases
 MONITOR program 165
 DISPLAY command 128, 251
 Divide by zero error 72
 Dollar sign 24
 Don't know response
 Affect on expressions 47
 Date question 43
 dBase 195
 Lotus 203
 MISSING keyword 75
 Number question 23
 Reset in free field data file 200
 Don't know responses
 Missing value code 91
 Short answer question 32

E

EDIT editor (DOS 5 or higher) 132
 Editing data
 See KEYPUNCH program
 Editing keys
 Startup Menu 125
 Editing records 169
 Epson printers 231
 EPSON.INI file 231, 234
 Equation processor 72
 Equations
 See Expressions
 Erase file 232
 Erasing interview records 85, 196
 ERR file extension 133, 261
 ERRORLEVEL batch command (DOS) 130
 Errors
 Beep 137, 155
 COMPILE program 132
 Display all compile errors 129
 Divide by zero 72
 Expression syntax 72
 Fatal errors 133

Finding with cross reference list 225
 Log file 133
 Rename log file 131
 Reporting software bugs 270
 Run-time math 72
 Warnings 133
 While running NCOLLECT program 159
 ESCAPE command 84, 117, 149, 246
 Escape key 84
 ESCAPE keyword 74, 117, 255
 Exclamation mark (!) 67
 EXIST command (DOS) 175
 Exiting questionnaires properly 84
 EXP function 79, 83, 257
 Expanded memory 5
 Exponentiation 255
 Exponents 73
 Expression keywords 255
 Expressions 72
 SHOWIF command 103
 Special keywords 74
 Types of questions used 73
 Extended IBM/PC character set 119
 Extended memory 5

F

FALSE keyword 74, 255
 Field names (askSam) 179, 194
 Field size
 Check-all-that-apply question 92
 Current date question 52, 96
 Current time question 54
 Date question 44, 96
 Fixed format file 92
 Match when copy from control file 108
 Multiple choice question 92
 Number question 25, 94
 Open-end question 95
 Random number question 92
 Short answer question 94
 Short answer questions 35
 Time question 97
 Version number question 59
 Version question 97
 Void question 60, 97
 File name extensions
 Summary list 261
 FILELIST.TXT 234
 FIX command 90, 243
 FLOOR function 79, 83, 257
 FOOTER command 146, 252
 FOREGROUND command 144, 173, 242, 253
 Foreign language error messages 252
 Foreign language surveys 119, 244
 Form feed 222
 FORMAT command 244, 249
 FORMAT command (DOS) 147, 175
 Formatted question list 220

Printing question names 223
 Titles and subtitles 223
 Free field data file
 Reset don't know response 200
 Variable names 200
 Free format data file 198
 Define date components 199
 Redefine string size 199
 Select records 188
 FRECUENCIAS command (SPSS) 219
 FROMJUL function 78-79, 256
 Functions 78

G

GDAY function 78, 256
 GET command 106, 251
 GJDAY function 82
 GMONTH function 78, 82, 256
 GOTO command 12, 241, 246
 GYEAR function 78, 82, 256

H

HEADER command 103, 146, 250, 253
 Help
 COLLECT command line options 139
 Command line option summary 258
 COMPILE command line options 132
 CONVERT command line options 182
 KEYPUNCH command line options 171
 MONITOR command line options 166
 NCOLLECT command line options 155
 Help screens
 See FOOTER command
 See HEADER command
 See Void question
 Hercules graphics card 128, 137, 154, 165
 Hewlett-Packard LaserJet Printers 231
 Hierarchy of arithmetic operators 73
 Hierarchy of operators 255
 HIGH command 253
 Change from version 1.0 244
 Date not known 47
 Date question 46
 Number question 27
 Short answer question 36
 HIGHLIGHT command 144, 173, 242, 253
 HPLJIICL.INI 234
 HPLJIICL.INI file 231
 HPLJIICP.INI 234
 HPLJIICP.INI file 231
 Hypertext 177, 193

I

IBM Proprinters 231
 IBMPRO.INI 234

IBMPRO.INI file 231
 IDENTIFY command 40, 112, 116, 250
 IF command 12, 69, 241, 246
 Rules 70
 Warning 257
 Importing control data files 118
 INCLUDE command 122, 126, 251
 INCREMENT command 117, 251
 Information screens
 See Void question
 INI file extension 230, 262-263
 Input range 27
 INSTALL.TXT 234
 Installation procedures
 Change from version 1.0 244
 Hard disk 235
 Programmer setup 234
 Interview call log 98
 Interview identification
 See Version number question
 INTRO.TXT 234

J

JDAY function 78, 81, 256
 JDAYOFWEEK function
 Return codes 80, 256
 JFHOLIDAY
 Return codes 81, 256
 JFHOLIDAY function 78, 80, 256
 JMONTH function 78, 81, 256
 JRN file extension 152, 262
 JYEAR function 78, 81, 256

K

KEYB command (DOS) 119
 KEYPUNCH program 169
 Back up data file 175
 Change from previous version 238-239
 Check current time and date 176
 Check free disk space 176
 Description 2
 DOS command line options 171, 259
 File name extensions 262
 Interviewer disk setup 174
 Password 172
 Print to file 171
 Printer initialization files 230
 Screen colors 171, 173
 Set workstation name 173
 Titles and subtitles 173
 KEYPUNCH.EXE 234
 Keyword summary 245
 Keywords 74
 KTX file extension 262

L

LAN 2
 LANGUAGE command 244
 LCD screens 128, 137, 154, 165
 LIB environment variable (DOS) 244
 Limits
 Answer lines 8
 Computer memory 5
 CONVERT file display 229
 Data translation rules 197
 dBase character fields 35, 195
 Decimal places 24
 Free disk space 148
 Left margin 223
 Lotus columns 201
 Lotus import files 199
 Lotus labels 35, 202
 Maximum card size 89
 Multiple choice answers 12
 Number of dBase fields 195
 Number of questions 5
 Number of SAS variables 212
 Number of SPSS/PC variables 218
 Password 172
 Password number 140, 155
 Programming numeric range checks 28
 Random number question 62
 SAS string fields 35
 SAS Variable labels 213
 Short answer field size 35
 Significant digits 25
 SPSS string fields 35
 SPSS variable labels 219
 Subtitles 174
 Text line length 8
 Title 174
 Variable names 7
 Line feed code 5, 91
 Line numbers
 Display when viewing text file 229
 LOG function 79, 83, 257
 LOG10 function 83
 Logic error 63
 LOGIC.ERR file 63, 261-262
 LOOK program
 DOS command line options 260
 LOOK.EXE 234
 Lotus 201
 /Print File command 118
 @DATE function 96, 202
 Display labels instead of numbers 203
 Don't know response 203
 European formatted numbers 120
 Importing free field files 199
 Labels 35
 Range names 202
 Select columns 186
 Select rows 188

Shorten long labels 202
 LOW command 243, 253
 Change from version 1.0 244
 Date not known 47
 Date question 46
 Number question 27
 Short answer question 36
 LPT1: 232

M

MAKEDEMO.PRG 234
 Managing telephone calls 99
 Master program disk 234
 Math
 See Expressions
 Math functions 79, 257
 MEANS command (SPSS/PC+ or Windows) 219
 Memory
 Saving by copying answers 29
 Merging program files
 See INCLUDE command
 Merging source programs 126
 MESSAGE.DOC 121, 234, 252
 MEWA.DOC 234
 MEWA.QPL 234
 MEWA2.DAT 234
 MEWA2.DOC 234
 MEWA2.QPL 234
 MID\$ function (BASIC) 206
 Minus sign
 Number field 94
 MISSING command 244, 249
 MISSING keyword 74, 255
 Missing value
 See Don't know responses
 Missing values 75, 91
 MOD function 79, 83, 257
 MONITOR program 150
 Description 2
 Don't display program title 166
 DOS command line options 165, 259
 MONITOR.EXE 234
 Monochrome graphics adapter 143
 Monochrome monitors 128
 MTX file name extension 168
 MULT keyword 247
 Multiple choice question 10
 Answer text lines 12
 COMPUTE command 14
 COMPUTEIF command 17
 Copy answer list 13
 Default answer 13
 Field size 92
 MULT keyword 11

N

NCOLLECT program 150
 Description 2
 Do not get workstation name 157
 Don't display program title 156
 Don't display question number 155
 Don't restart interview 156
 DOS command line options 154, 259
 Error beep 155
 File name extensions 262
 Limit control file display 154
 Network setup issues 158
 Novell NetWare 150
 Password 155
 Set retry time 157
 Set workstation name 156
 Uppercase option 156
 Use first free control file case 154
 Using workstation name 33
 NCOLLECT.EXE 234
 Negative numbers 25, 243
 Range limits 28
 Network 2
 Copying the workstation name 33
 Software installation 236
 System interrupts 157
 Network requirements 150
 Network setup issues 158
 NEW keyword 74-75, 239, 255
 NEWCARD command 90, 249
 NEXT command 7, 12, 65, 246
 Novell
 Capture user account name 156
 NUMBER keyword 247
 Number question
 Copying answer line 29
 Default answer line 25
 European format 119
 Field size 94
 Formats 24
 Input range 27
 Limiting input range 27
 NUMBER keyword 24
 Setting decimal places 24
 Number questions 23

O

Okidata printers 231
 OKIDATA.INI 234
 OKIDATA.INI file 231
 Open-end question
 Edit buffer size 41
 Field size 95
 Open-end questions
 Monitoring free disk space 148, 176
 Open-ended question 39
 Open-ended questions

analyzing results with askSam 177, 193
 Operators 72
 OPTION keyword 74, 77, 239, 255
 COLLECT command line option 139
 KEYPUNCH command line option 172
 NCOLLECT command line option 155
 OR operator 255
 Order of evaluation 73
 OTX file extension 152, 261-262

P

Page
 SAS 214
 SPSS 220
 Page break
 Do not break pages 223
 See PAGE command
 PAGE command 224, 254
 Page size 221
 Left margin 222
 Length 222
 Width 221
 Parentheses 73, 255
 Passing information from the command line
 See OPTION keyword 77
 PASSWORD command 131, 253
 PASSWORD keyword 74, 255
 Password option 140, 155, 172
 PATH command (DOS) 235
 PAUSE command 116, 251
 Period
 See Command lines
 Permission denied 160
 PRINT command (DOS) 220
 Printer control codes 230
 Printer initialization file 171, 183
 Printing text files 229
 PRN file name extension 198, 263
 PROC FREQ (SAS) 213
 PROC MEANS (SAS) 214
 PROC PRINT (SAS) 214
 Program generation
 See CONVERT program
 Program lines 5, 245
 PUBLISH program 241
 See CONVERT program
 See Formatted question list
 PUBLISH.EXE
 See CONVERT program
 PUT command 108, 251

Q

Q1.DOC 234
 Q1.QPL 234
 Q2.DOC 234
 Q2.QPL 234
 QBasic interpreter 206

QPL data file 88
 QPL file extension 124, 136, 261
 QPL40.EXE 236
 QPLLIB.EXE 244
 QPLPRINT.INI file 172, 183, 230, 262
 QUESTION command 6, 246
 Question names
 See Variable names
 Questionnaire
 See Formatted question list
 Questions
 Check-all-that-apply 18
 Current date 51
 Current time 53
 Date 43
 Maximum number of 5
 Multiple choice 10
 Naming 7
 Number 23
 Open-ended 39
 Random number 61
 References to answers 8
 Select for analysis 186
 Short answer or string 32
 Syntax 4
 Text lines 8, 53
 Version number 58
 Void 60

R

RANDOM keyword 248
 Random number question 61
 Field size 92
 RANDOM keyword 61
 README.TXT 234
 RECORD keyword 74, 189, 255
 Record layout
 See Cross-reference list
 Error in Ver. 2.0 242
 Fixed format data 88
 Matching an existing data file 90
 Printed on formatted question list 223
 Translate data to new 196
 Translation to new layout 3
 Records
 Append 196
 Copy 195
 Delete 196
 Select for analysis 188
 RELOADIF command 112, 250
 Rename file 232
 Respondent database 99
 Restarting interviews at point of interruption 116
 Root directory
 Maximum number of files 43
 ROUND function 79, 82, 257

S

SAS 208
 Data file 88
 Data formats 92
 Dates 213
 Descriptive statistics 214
 European formatted numbers 120
 Frequency statistics 213
 List string questions 214
 Missing values 91
 Number of variables 212
 Page size 214
 SAS/PC or Windows 208
 Select variables 186, 212
 Shorten string variables 212
 String fields 35
 Variable labels 213
 View program listing 229
 SAS file extension 263
 SAVECONTROLIF command 110, 117, 242, 251
 SAVEIF command 85, 242, 246
 Saving valid questionnaire responses 85
 Scheduling interviews 98
 Automated call log 99
 Scientific notation 25
 SELECT IF statement (SPSS) 87
 SGN function 79, 82, 257
 Shift key 32
 Short answer question 32
 analyzing results with askSam 177, 193
 Default answer 35
 Field size 94
 Formats 33
 Limiting range of answers 36
 Short answer questions
 STRING keyword 33
 SHOWIF command 102, 112, 117, 250
 Use with PUT command 109
 SKIP command 66
 Skip instructions 12, 65
 See IF, GOTO, and NEXT commands
 See NEXT command
 Random number question 62
 Skip path 65, 71, 257
 Coding for skipped questions 91
 Source questionnaire program 126
 Display during compile 128
 DOS text file 1, 5
 Program lines 5
 Syntax 4
 SPANISH.DOC 123, 234, 252
 SPC file extension 263
 Spreadsheet
 See Lotus
 SPS file extension 263
 SPSS 215
 Data file 88

Dates 218
 Descriptive statistics 219
 European formatted numbers 120
 Frequency statistics 219
 List string questions 220
 Maximum card size 89
 Missing values 91
 Number of variables 218
 Page size 220
 Redefine long strings 218
 SELECT IF statement 87
 Select variables 186, 217
 SPSS for Windows 215
 SPSS-x 215
 SPSS/PC+ 215
 String fields 35
 Value labels 92
 Variable labels 219
 View program listing 229
 SPX file extension 263
 SQR function 79, 84, 257
 SQUASH.EXE 161, 236, 260
 Startup menu
 Editing keys 124
 Statistics
 Descriptive 214, 219
 Frequency 213, 219
 Generating analysis programs 3
 String
 See Short answer question
 STRING keyword 248
 SUBTITLE command 174, 223, 254
 Syntax
 Source program 4
 Syntax changes
 Ver. 1.0 to 3.0 244
 Ver. 2.0 to 3.0 240

T

Tab delimited data files 198
 TAWK compiler 204
 TEST1.DOC 234
 TEST1.QPL 234
 TEST1.TXT 234
 Text In/Out (WordPerfect) 220
 Questions 6
 Thousands separator
 European number format 120
 Time
 Check for correct on startup 148
 Correct MONITOR display time 164
 TIME command (DOS) 148, 175
 TIME keyword 53, 74, 248, 255
 When not to use 71
 Time of interview
 See Current time question
 Time question
 See Current time question

Field size 97
 Use instead of TIME keyword 71
 Time, display elapsed time 145
 Timing a question 54
 TITLE command 173, 223, 254
 TOFREE.EXE
 See CONVERT program
 TOJUL function 78-79, 256
 TONEW.EXE
 See CONVERT program
 TOSAS.EXE
 CONVERT program 244
 TOSPSS.EXE
 See CONVERT program
 Training interviewers 220
 TRUE keyword 74, 255
 TXT file extension 220, 263
 TYPE command 6, 247

U

Unconditional skips 65
 Underlining 5
 UPPERCASE command 37, 141, 242, 253
 Uppercase option 141, 156

V

Variable names 7
 See Declaring at QUESTION command
 Defined in Lotus 202
 Fixed format data 88
 Include in free field file 200
 Including on formatted question list 223
 References to answers 8
 SAS 212
 SAS labels 213
 Select for analysis 186
 SPSS labels 219
 SPSS program 217
 Use in expressions 74
 VERBOSE command 129, 251
 VERSION keyword 249
 Version number question 58
 Default answer 59
 Field size 59
 VERSION keyword 59
 Version question
 Field size 97
 VOID keyword 249
 Void question 60
 Field size 60, 97
 VOID keyword 60

W

Warnings 133
 Windows Program Manager 151

WK1 file extension 263
 WKS file extension 263
 Word processor 1
 Word-wrap 5
 Word-wrapping 222
 WordPerfect 5
 Workstation ID already in use 161
 Workstation name
 Short answer questions 33
 WORKSTATION subargument 240
 WRS file extension 263

X

XDATE keyword 51, 249
 XRF file extension 263

Y

YRMODEA function (SPSS) 218
 YYMMDD and YYYYMMDD date formats 44
 YYMMDD date format 51, 119, 199, 208, 213, 218

Comments

Please send any comments or suggestions you have about the software or manuals to:

Kevin Dooley
General Government Division
U.S. General Accounting Office
441 G Street, N.W.
Washington, D.C. 20548

Email: dooleyk.ggd@gao.gov